



[12] 发明专利说明书

专利号 ZL 200710137398.3

[45] 授权公告日 2010年3月17日

[11] 授权公告号 CN 100594470C

[22] 申请日 1998.9.24

[21] 申请号 200710137398.3

分案原申请号 98809472.X

[30] 优先权

[32] 1997.9.25 [33] US [31] 60/060,223

[32] 1998.9.24 [33] US [31] 09/160,433

[73] 专利权人 蒂吉通信系统公司

地址 美国华盛顿

[72] 发明人 爱德华·P·弗林切姆

戴尔·L·格罗夫

切里尔·格朗伯克 马丁·T·金

克里福德·A·库什勒

[56] 参考文献

EP0660216A1 1995.6.28

US4891786A 1990.1.2

CN2147580Y 1993.11.24

审查员 杨 洁

[74] 专利代理机构 中国国际贸易促进委员会专利  
商标事务所

代理人 蒋世迅

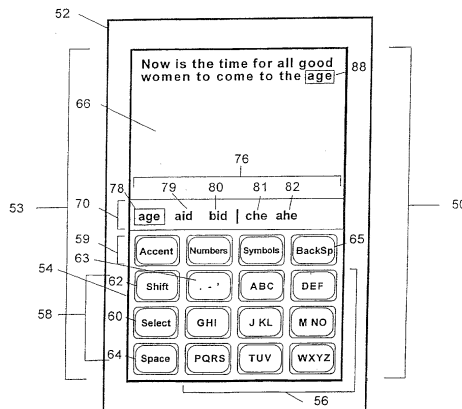
权利要求书 8 页 说明书 41 页 附图 33 页

[54] 发明名称

用于对用户输入的多义性输入序列进行多义性消除的系统和方法

[57] 摘要

一种简化键盘多义性消除系统。键盘有 12 - 16 个键，9 个键带数字字母和其他符号，并且该 9 个键合另外 1 个用作 10 个数字键。文本输入键是有多义性的。用户利用选择键或其他无多义性键限定每个单词输入的结束而可将输入的键击序列与多个具有同样的键击序列的单词匹配。输入的每键击序列经过全部词汇对比处理而选择出匹配的并一使用频率的降序显示给用户。以特殊格式存储的词汇支持压缩而不会增加处理时间。起始字母匹配的长单词的词干也显示以减小使用频率。如与词干相关联的单词使用频率高就作为默认单词对象显示。激活与多个标点相关联的多义性键可根据上下文减小多义性消除处理。



1. 一种用于对用户输入的多义性输入序列进行多义性消除的多义性消除系统，此多义性消除系统的构成包括：

(a) 具有多个输入的用户输入装置，多个输入的每个都与多个字符相关联，输入序列是在每次通过操作用户输入装置进行选择输入时生成的，其中生成的输入序列对应于被选定的输入序列，并且其中生成的输入序列具有文本解释，该文本解释由于与每个输入相关联的多个字符而具有多义性；

(b) 包含多个对象的存储器，多个对象的每个都与输入序列相关联，其中存储器中的多个对象中的每个还都与使用频率相关联；

(c) 为用户显示系统输出的显示器；及

(d) 与用户输入装置，存储器和显示器连接的处理器，所述处理器的构成包括：

(i) 用于从存储器内的多个对象中认定至少一个与每个生成的输入序列相关联并具有最高使用频率的对象的认定部件；

(ii) 用于显示至少一个认定的与每个生成的输入序列相关联的对象作为生成的输入序列的文本解释的输出部件，其中与每个输入序列相关联的对象以使用频率的降序被显示；以及

(iii) 用于在检测到用户输入装置的一个或多个无多义性选择输入操作时选择认定对象中的一个输入到文本输入显示位置的选择部件；其中所述多义性消除系统的特征在于：

至少多个输入中的一个与多个标点字符相关联的多义性标点输入，并且其中处理器，在检测到选择多义性标点输入并将其添加到输入序列时，临时输出终止于相关联的标点字符之一的输入序列的相关联的文本解释，并且在检测到选择下一个输入时，其中当所述下一个输入是指定输入的第一集合的元素，其中所述指定输入的第一集合的每个都是与下一个同临时输出的标点字符不同的字符的输出相关联的，肯定前面输出的标点字符，其中当所述下一个输入

是指定输入的一个或多个集合的元素时，其中所述指定输入的一个或多个替代集合的每个都是与下一个同所述临时输出的标点字符不同的字符的输出相关联的，利用与多义性标点输入相关联的多个标点字符的另一个代替前面输出的标点字符，其中的代替标点字符是由指定输入的替代集合确定的，其中所述下一个输入是其一个元素。

2. 如权利要求 1 的系统，其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与包含添加的多义性标点输入的所述输入序列相关联的对象而确定的。

3. 如权利要求 1 的系统，其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与包含添加的多义性标点输入并且还在所述下一个输入是多义性输入时添加所述下一个输入的所述输入序列相关联的对象而确定的。

4. 如权利要求 1 的系统，其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与以选择多义性标点输入开始并在所述下一个输入是多义性输入时添加所述下一个输入而确定的。

5. 如权利要求 1 的系统，其中多义性标点输入与句点，短划及撇号相关联。

6. 如权利要求 2 的系统，其中处理器，在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成不与存储器中的任何对应于完整单词的对象相关联的序列时，临时输出一个相关联的标点字符，并且

其中处理器在检测到选择下一个输入时，其中所述下一个输入是与一个或多个字母相关联但不与字母“s”相关联的多义性输入，将前面输出的标点字符改变为短划，在选择多义性标点输入之前肯定前面临时输出的当前输出的输入序列的文本解释，并开始新生成的输入序列，该新生成的输入序列包括所述下一个多义性输入，

其中处理器在检测到选择下一个输入,其中所述下一个输入是与一个或多个包含字母“s”的字母相关联的多义性输入时,在选择多义性标点输入之前肯定前面输出的当前生成的输入序列的文本解释,开始新生成的其构成包括所述与一个或多个包含字母“s”的字母相关联的下一个多义性输入的输入序列并以撇号替代前面输出的标点字符和临时添加字母“s”作为所述新生成的输入序列的文本解释,并且其中处理器在检测到选择另外一个下一个输入,其中所述下一个输入是与一个或多个字母相关联的多义性输入时,在选择多义性标点输入之前以短划替代前面输出的撇号,并开始新生成的其构成包括与一个或多个包含字母“s”的字母相关联的后面跟着与一个或多个字母相关联的另外一个下一个多义性输入的输入序列,并输出存储器中的与所述新生成的输入序列相关联的对象,所述新生成的输入序列包括与一个或多个包含字母“s”的字母相关联的后面跟着与一个或多个字母相关联的另外一个下一个多义性输入的所述多义性输入,并且

其中处理器在检测到选择下一个输入时,其中所述下一个输入不与一个或多个字母相关联,肯定前面输出的标点字符。

7.如权利要求4的系统,其中处理器,在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成不与存储器中的任何对象相关联的序列时,接受在检测到选择多义性标点输入之前临时显示的具有最高使用频率的与生成的输入效率相关联的对象,临时输出一个默认的与多义性标点输入相关联的标点字符,并且建立一个长度为一个输入的新生成的输入序列,该一个输入包括多义性标点输入。

8.如权利要求1的系统,其中处理器,在检测到在生成的输入序列中出现选择多义性标点输入时生成一个可由用户通过在用户输入装置上一次或多次操作无多义性选择输入而选择的替代的文本解释的序列,其中所述的替代的文本解释的序列的第一个的生成是通过将一个与多义性标点输入相关联的标点字符添加到包括了与在多义

性标点输入之前输入的生成的输入序列中的输入序列相关联的具有最高使用频率的对象的第一对象上，其后是添加第二对象，所述第二对象包括了与在多义性标点输入之后输入的生成的输入序列中的输入序列相关联的具有最高使用频率的对象，其后是与生成的输入序列中的先于多义性标点输入输入的输入序列相关联的包含具有越来越低的使用频率的对象的其他对象组合，以及与生成的输入序列中的后于多义性标点输入输入的输入序列相关联的具有越来越低的使用频率的对象，并且使生成的这些对象以所述第一和第二对象的混合频率次序的降序生成。

9. 如权利要求 1 的系统，其中处理器，在检测到选择一个或多个后面为第一选择的无多义性选择时，选择具有最高使用频率的对象并向显示器输出所述具有最高使用频率的对象及单个后面空格，并且其中在检测到每个后面重复的连续的无多义性选择输入时从认定的对象中选择具有下一个较低使用频率的对象，并且在先于前面输出的单个空格的其显示位置以每个具有认定的对象中的下一个较低使用频率的所述对象替代前面输出的文本对象。

10. 如权利要求 2 的系统，其中处理器，在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成与存储器中的具有最高使用频率的结尾为与所述多义性标点输入相关联的完整的单词对象相关联的序列时，并且在检测到下一个输入的选择，其中所述的下一个输入是与一个或多个字母相关联的多义性输入时，肯定所述完整的单词对象的输出是当前生成的包含多义性标点输入的选择，并开始一个新生成的构成与一个或多个字母相关联的所述下一个多义性输入的输入序列。

11. 一种用于对用户输入的多义性输入序列进行多义性消除的方法，此方法的构成包括：

(a) 由多个输入生成输入序列，多个输入中的每个与多个字符相关联，生成的输入序列具有的文本解释由于与每个输入相关联的多个字符而具有多义性；

(b) 存储多个对象，多个对象的每个都包括与输入序列和使用频率相关联的字符序列；

(c) 从多个对象中认定至少一个与每个生成的输入序列相关联并具有最高使用频率的对象；

(d) 显示至少一个认定的具有最高使用频率的与每个输入序列相关联的对象作为生成的输入序列的文本解释，其中与每个生成的输入序列相关联的对象以使用频率的降序被显示；以及

(e) 在检测到一个或多个无多义性选择输入的选择时选择认定对象中的一个作为生成的输入序列的文本解释；其中多个输入中的至少一个是与多个标点字符相关联的多义性标点输入，本方法的特征在于包括如下步骤：

(a) 检测包含多义性标点输入的输入序列的生成；

(b) 在检测到选择所述多义性标点输入时，临时输出一个相关联的标点字符；

(c) 在检测到选择下一个输入时，其中所述下一个输入是指定输入的第一集合的元素，并且指定输入的所述第一集合中的每一个与不同于所述临时输出的所述相关联的标点字符之一的下一个字符的输出相关联，肯定前面输出的标点字符；

(d) 在检测到选择所述下一个输入时，其中所述下一个输入是指定输入的一个或多个替代集合之一的元素，其中所述指定输入的一个或多个替代集合的每个都是与下一个同所述临时输出的所述相关联的标点字符之一不同的字符的输出相关联的，利用与多义性标点输入相关联的多个标点字符的另一个代替前面输出的标点字符，其中的代替标点字符是由指定输入的替代集合确定的，其中所述下一个输入是其一个元素。

12. 如权利要求 11 的方法，其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与包含添加的多义性标点输入的所述输入序列相关联的对象而确定的。

13. 如权利要求 11 的方法, 其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与包含添加的多义性标点输入并且还在所述下一个输入是多义性输入时添加所述下一个输入的所述输入序列相关联的对象而确定的, 该多义性输入是所述第一集合或所述指定输入的一个或多个替代集合的元素。

14. 如权利要求 11 的方法, 其中所述第一集合和所述指定输入的一个或多个替代集合的组成是根据在所述存储器中是否存在一个或多个与以选择多义性标点输入开始并在所述下一个输入是多义性输入时添加所述下一个输入而确定的, 该多义性输入是所述第一集合或所述指定输入的一个或多个替代集合的元素。

15. 如权利要求 12 的方法, 还包括:

(a) 在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成不与存储器中的任何对象相关联的序列时, 临时输出一个相关联的标点字符; 以及

(b) 在检测到选择下一个输入时, 其中所述下一个输入是与一个或多个字母相关联但不与字母“s”相关联的多义性输入,

(i) 将前面输出的标点字符改变成短划;

(ii) 在选择与一个或多个字母相关联但不与字母“s”相关联的多义性标点输入之前肯定当前生成的输入序列的前面输出的文本解释;

(iii) 开始新生成的输入序列, 其构成包括与一个或多个字母相关联但不与字母“s”相关联的所述下一个多义性输入; 以及

(c) 在检测到选择下一个输入, 其中所述下一个输入是与一个或多个包含字母“s”的字母相关联的多义性输入时,

(i) 在选择多义性标点输入之前肯定前面输出的当前生成的输入序列的文本解释;

(ii) 开始新生成的输入序列, 其构成包括与一个或多个包含字母“s”的字母相关联的所述下一个多义性输入

(iii) 以撇号替代前面输出的标点字符和临时添加字母“s”作为所述新生成的输入序列的文本解释；以及

(iv) 在检测到选择另外一个下一个输入，其中所述下一个输入是与一个或多个字母相关联的多义性输入时，以短划替代前面输出的撇号，并开始新生成的在所述短划之后的其构成包括与一个或多个包含字母“s”的字母相关联的后面跟着与一个或多个字母相关联的另外一个下一个多义性输入的输入序列，并输出存储器中的与所述新生成的在所述短划之后的输入序列相关联的对象，并且

(d) 在检测到选择下一个输入时，其中所述下一个输入不与一个或多个字母相关联，肯定前面输出的标点字符。

16. 如权利要求 14 的方法，还包括：

在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成不与存储器中的任何对象相关联的序列时，

(a) 在检测到选择多义性标点输入之前接受临时显示的与生成的输入序列相关联的具有最高使用频率的对象；

(b) 临时输出一个与多义性标点输入相关联的默认标点字符；以及

(c) 建立一个长度为一个输入的新生成的输入序列，该一个输入包括多义性标点输入。

17. 如权利要求 11 的方法，还包括：

在检测到在生成的输入序列中出现选择多义性标点输入时，

(a) 生成一个可由用户通过在用户输入装置上一次或多次操作无多义性选择输入而选择的替代的文本解释的序列，其中所述的替代的文本解释的序列的第一个的生成是通过将一个与多义性标点输入相关联的标点字符添加到包括了与在多义性标点输入之前输入的生成的输入序列中的输入序列相关联的具有最高使用频率的对象的第一个对象上，其后是添加第二对象，所述第二对象包括了与在多义性标点输入之后输入的生成的输入序列中的输入序列相关联的具有



最高使用频率的对象，其后是与生成的输入序列中的先于多义性标点输入输入的输入序列相关联的包含具有越来越低的使用频率的对象的其他对象组合，以及与生成的输入序列中的后于多义性标点输入输入的输入序列相关联的具有越来越低的使用频率的对象，并且使生成的这些对象以所述第一和第二对象的混合频率次序的降序生成。

(b) 在检测到用户输入装置的一次或多次的无多义性选择输入操作时选择所述的替代的文本解释的序列中的一个文本解释。

18. 如权利要求 11 的方法，还包括：

(a) 在检测到选择一个或多个后面为无多义性选择的第一选择时，选择具有最高使用频率的对象并输出所述选择的具有最高使用频率的文本对象及单个后面空格；以及

(b) 在检测到每个后面重复的连续的无多义性选择输入时，

(i) 从认定的对象中选择具有下一个较低使用频率的对象；并且

(ii) 在先于前面输出的单个空格的其显示位置以每个具有下一个较低使用频率的认定的对象中的文本选择对象替代前面输出的文本对象。

19. 如权利要求 12 的方法，还包括：

在检测到选择多义性标点输入并将所述多义性标点输入添加到当前生成的输入序列以形成与存储器中的具有最高使用频率的结尾为与所述多义性标点输入相关联的完整的单词对象相关联的序列时，并且在检测到下一个输入的选择，其中所述的下一个输入是与一个或多个字母相关联的多义性输入时，

(a) 肯定所述完整的单词对象的输出是当前生成的包含多义性标点输入的输入序列的文本解释；以及

(b) 开始一个新生成的构成与一个或多个字母相关联的所述下一个多义性输入的输入序列。

## 用于对用户输入的多义性 输入序列进行多义性消除的系统和方法

本申请是中国专利申请号为 98809472.X、申请日为 1998 年 9 月 24 日、发明名称为“简化键盘多义性消除系统”一案的分案申请。

### 技术领域

本发明一般涉及简化键盘系统，并且更具体涉及利用单词级多义性消除来解决多义性键击的简化键盘系统。

### 背景技术

多年来，便携式计算机变得越来越小。在生产更小的便携式计算机的尝试中限制尺寸的主要部件一直是键盘。如采用标准打字机尺寸的操作键，则便携式计算机一定得至少与键盘一般大。在便携式计算机上一直使用小型键盘，不过小型键盘的操作键太小，用户操作不便，速度也慢。

在便携式计算机中采用全尺寸键盘也会阻碍计算机真正用作便携式计算机。多数便携式计算机不放置于平整的工作表面上使用户能够双手打字就无法使用。用户站立或移动时也不易使用便携式计算机。在称为个人数字助理(PDA)的最新一代的小型便携式计算机中各公司企图通过在 PDA 中采用手写识别软件解决这个问题。用户可直接在触敏板或触敏屏上书写来输入文本。之后利用识别软件将这种手写文本转换为数字数据。很不幸，除了一般用笔书写比打字慢以外，手写识别软件的精确性和速度至今远不能令人满意。更坏的是如今的要求输入文本的手持计算装置还在越变越小。在双向寻呼，手机及其他便携无线技术中的最新进步要求小型便携式双向信息系统，并且特别是要求可收发电子函件(“e-mail”)的系统。

因此,如果开发一种键盘使用户能够在用一只手拿着小型计算机的同时使用另一只手操作向其中输入文本将是很有益处的。现有的开发工作一直考虑采用减少键数的键盘。正如按键式电话机的键座布局所建议的,很多简化键盘采用 3×4 阵列的排键方式。按键阵列中的每一个键包含多个字符。于是在用户输入一个按键序列时就会出现多义性,因为每一个按键代表数个字母。曾经提出过数种方案欲解决键击序列的多义性问题。

提出的一种无多义性指定简化键盘输入字符的方案要求用户以两次以上键击来指定每个字母。键击可以同时输入(联合键击)或顺序输入(多次键击指定)。无论是联合键击还是多次键击指定都产生了一种足够简单并且使用有效的键盘。多次键击指定效率低,而联合键击复杂,不便记忆和使用。

提出的另外一种判断对应于多义性键击序列的正确字符序列的方案总结于下面的文章中:篇名为“Probabilistic Character Disambiguation for Reduced keyboard Using Small Text Samples”,著者为 John L. Arnott 及 Muhammad Y. Javad,发表在 Journal of the International Society for Augmentative and Alternative Communication 中(以下称之为“Arnott 一文”)。Arnott 一文指出大多数多义性消除方案应用相关语言中字符序列的公知统计数据来解决给定文境中的字符多义性。即现存的多义性消除系统是从统计上分析用户输入的多义性键击组合以求判断键击的恰当解释。Arnott 一文还指出,有几种多义性消除系统企图使用单词级多义性消除来解码由简化键盘输入的文本。单词级多义性消除是通过在接收到一个表示单词结束的无多义性字符之后将所接收到的键击序列与字典中的可能匹配进行比较而进行多义性消除。Arnott 的文章讨论了单词级多义性消除的很多缺点。例如,经常不能正确地解码单词,因为在辨识非常见单词受到限制,并且无能力解码字典中未包含的单词。由于解码的限制,单词级多义性消除不能以一键一字符的效率对无限制英文文本产生无错误解码。因此,Arnott 一文的重点放在字符级多义性消除,而不是单词级多义性消除,并指出字符级多义性消除似乎是最有希望的多义性消除技术。

一种建议的基于单词级的多义性消除的方案公开在标题为“Principles of Computer Speech”的一本教科书中，著者是 I. H. Witten，出版者为 Academic Press (1982) (以下称之为“Witten 方案”)。Witten 讨论的是一种减小利用电话按键键座输入的文本多义性的系统。Witten 承认对于一本 24500 个单词的字典中的大约 92% 的单词在将键击序列与字典进行比较时不会出现多义性。然而，Witten 指出，当多义性的确出现时，多义性必须由系统将多义性提交给用户并要求用户在多个多义性条目中进行选择以交互方式来解决。因此，用户必须对系统在每一个单词末尾的预测做出反应。这种反应降低系统的效率并增加为输入给定的文本段所需键击数。

已经投入很大的努力来减少输入文本段所需要的键击的数目以求达到为了使多义性消除键盘可被接受用于便携式计算机的必需效率。单词级多义性消除的任何应用面对的最大挑战是能否成功地将其应用于对其应用最为有利的各种硬件平台上。如上所述，此类装置包括双向寻呼，手机及其他手持无线通讯装置。这些系统使用电池作为电源，因此其硬件设计和资源利用都尽可能节约。设计为在此种系统上运行的应用程序必须将处理器带宽的使用及存储器要求减小到最低。这两个因素的关系一般是相反的。因为单词级多义性消除系统要求有很大的单词数据库运行，并且必须能迅速对输入的键击发出反应并提供令人满意的用户界面。另外一个挑战是压缩所需数据库而不致严重影响利用数据库所需的处理时间。

单词级多义性消除应用面对的另一个挑战是要向用户提供有关输入的键击的足够的反馈。在使用普通的打字机或字处理器时每个键击都代表一个可以在输入后尽快向用户显示的唯一字符。可是在单词级多义性消除中这一点经常是不可能做到的，因为每个键击代表多个字符，并且任何键击序列都可能与多个单词或单词词干匹配。特别是在用户拼写或键击出错时这就更加成为问题，因为用户没有把握判断是否出错，必须一直等到整个键击序列输入完毕而所要求的单词未出现才知道。

针对上述挑战的单词级多义性消除系统公开在 PCT 申请 PCT/

US96 / 12291 中，标题为“Reduced Disambiguating System”，发表日期为2月13日，是作为 International Publication(国际出版物) No. WO 97/05541(以下称之为 WO 97/05541 出版物)。

公开在 WO 97/05541 中的系统包含：一个键盘，其中多个字母和符号被赋予至少几个键；一个存储器，存储一个或多个词汇模块；一个显示器，其中呈现系统文本输出；及一个处理器，有效连接到键盘、存储器和显示器。单词和单词词干（对应于一个单词的开始字母序列）被使用含有多个互连节点的树形结构存储在词汇模块中。

在公开于此 WO 97/05541 中的单词级多义性消除系统工作期间，当用户输入一个键击序列时，系统处理器访问系统存储器以构造于输入的键击序列相对应的单词和单词词干。特别是每个输入的键击序列对应词汇模块树形结构的一个节点，该节点反过来又对应一个单词或单词词干。当输入附加键时，系统处理器通过将一组与先前的键击序列（即不包含最后键击的键击序列）相关联的单词和/或单词词干结合而形成新的一组一个或多个单词和/或单词词干。在存储器树形结构的环境下，一组一个或多个单词词干和/或候补单词是通过访问与树结构的节点相关联的单词和/或单词词干而派生的，这些节点与代表先行键击序列，并且，另外，还代表对应于整个键击序列的可能的单词词干和单词的节点互相连接。

根据公开在 WO 97/05541 中的装置，单词和单词词干是根据与被显示的单词和单词词干相关联的频率显示于系统显示装置上。当候补单词列表显示时，系统用户操作指定的键盘键来选择所要求的候补单词。

### 发明内容

本发明提供一个利用单词级多义性消除来解决键击中的多义性的简化键盘。在一个实施例中，系统包含一个触敏显示板，其中与显示器表面接触可对系统生成对应于接触地点的输入信号。另一种

代替方式是键盘可采用全尺寸的机械键。

在一个优选实施例中，9个符号和字母键被分组成3×3阵列，另带3至6个附加的特殊功能键。某些键被赋予多个字母和符号，结果这些键(下面称之为“数据键”)的键击是有多义性的。用户可输入一个其中每一个键击对应于单词中的一个字母的键序列。因为单个键击有多义性，所以键击序列潜在地具有以同样数目的字母与一个以上的单词相匹配的能力。键击序列由将此序列与相应的存储单词或其他解释进行匹配的词汇模块实施处理。在接收到每一键击时在显示器上以选择列表的形式将与键击序列匹配的单词及单词词干提供给用户。

根据本发明的一个方面，键中有一个被赋予多个标点字符。在一个优选实施例中，句点“.”，短划“-”，和撇号“'”赋予一个键。用户可在需要加入标点的地方通过一次激活一个键打入该键上的字符。因为多个标点字符赋予了一个键，所以该键击有多义性。本发明的系统可从前后的键击判断要打入的是哪一个标点并自动生成该标点。此键击序列的替代解释也可以选择列表的形式提供给用户。

最好是单词的解释以使用频率递减的顺序列出，最常用的单词列在第一个。在选择列表中的项目可通过按下一次或多次“选择”键而选定。可通过按下“退格”键“取消”键击。

用户可通过按下“选择”键来限定输入的键击序列。在接收到“选择”键之后，多义性消除系统选择最常用的单词并将此单词添加到正在构造的句子中。“选择”键是用来限制输入的键击序列。单独的显示“空格”键是用来生成空格字符，也用来限制输入的键击序列。在一个优选实施例中，“选择”键还“额外负担”第二种功能，该功能为该“选择”键还在所选择的单词的后面生成一个空格，即简化键盘多义性消除系统自动在单词中间插入一个合适的空格。

“选择”键还用来从提供给用户的选择列表中选择不大常用的单词。如在提供给用户的选择列表的顶端的单词不是所要求的单词，

用户可再一次按下“选择”键从第一最常用单词移到第二个最常用的单词，以及再一次移到到第三个最常用的单词，依此类推。简化键盘多义性消除系统的此实施例没有在条目被选择时可作用于条目的专用“执行”键或“接受”键。一旦用户选定一个单词，则在接收到下一个符号或字符键击时，该单词就自动地被“接受”输入并添加到正在组合的句子。

用来进行键击多义性消除的语言学数据库包含容许系统将下面的单词或单词词干对象作为第一对象提出的信息，该单词或单词词干对象，虽然不是与对应于此当前键击序列的节点相关的最常用的单词，却形成一个更长单词或总频率大于与当前节点相关的最常用的单词的频率的多个单词的词干。此“词干提升(stem promotion)”特点特别适用于显示器面积有限的系统，因为选择列表实际上不能显示于屏幕上，用户在屏幕上只能看到在插入点处显示的作为选择列表上第一个对象的单词。结果该对象作为一个整体而言在从一个键击到另一个键击时其变化不太频繁，界面不大“跳动”并因而不大令人分心和迷惑。

对象，如单词和单词词干，是存储于使用树形数据结构的词汇模块中。在此安排中，与具体键击序列相对应的单词是利用与紧前方的键击序列(即缺少最后键击的具体键击序列)相关联的一组单词或单词词干构造而成的。以这种方式构造单词可减少词汇模块的存储空间，因为单词词干只存储一次，在树形结构的顶端，并且为利用它们构造的所有的单词共享。树形结构也可大大减少处理要求，因为无须通过检索来定位所存储的对象。存储于树形数据结构中的单词或单词词干可包含频率或其他指示将哪一个项目第一个提供给用户的排序信息，从而可进一步减少处理要求。在本发明的实践中，此树形数据结构要利用专门的算法进行改造以便进一步压缩数据库所需的总容量而又不会在利用其进行检索与键击序列相关联的对象时造成额外的处理负担。经过改造的树形结构的另外一个方面是可自动认定用于将对象与键击序列相联系的通用化规则。利用此种规

则，词汇模块可以以很高的成功几率将键击序列与起初生成该模块时未曾使用的单词与单词词干相关联。

在优选实施例中的键的内部、逻辑表示不需要与由实际键上的键标所表示的物理配置完全相同。比如，在构造用来表示法文词汇模块的数据库中，三个重音符(ÂÇÀ)也可与不带重音符的字符 ABC 相关联。这就使用户可以想起并打入包含特定的带重音符的字符的单词，为此只需要对每个字符执行一次键激活，简单地只要针对相关的带重音符的字符 (accented character) 激活逻辑上与其相关联的物理键即可。

将多个字母赋予一个键的组合效应，利用“选择”键限制单词，在选择列表中将最常出现的单词或单词词干列为第一个单词，在选择列表中插入多个解释，自动通过后面单词的第一个键击将选定的单词添加到句子，自动添加空格，可将用于多义性消除的大型数据库压缩而不会招致任何严重的处理损失的能力，可通过键入与不带重音符的字母相关联的键生成带重音符的特殊字符的能力，以及可根据键击的上下文自动在赋予一个键的多个标点字符之间进行多义性消除的能力产生令人惊奇的结果：对于多种语言，在具有代表性的文本材料语料库中的远远超过 99%的单词可以以极高的效率在此系统上键入。采用简化键盘多义性消除系统输入这些单词中的大致 95%所要求的输入键击数与利用通常的键盘输入的键击数相同。当单词包含重音符时，单词输入所需的键击数比采用通常键盘时的键击数为少。当单词是以使用频率排序时，所要求的单词通常都是排列在第一个，并且经常是列出的唯一一个。于是用户无须利用多于通常的键击数的键击而进行下一个单词的输入。因而就可以利用具有数量少的全尺寸键的键盘进行文本的高速输入。

此处所公开的简化键盘多义性消除系统可使采用本系统的计算机或其他装置的尺寸减小。键的数目减少就使得可以将装置构造成可由用户以一只手拿持，而用另外一只手操作。所公开的系统特别适用于 PDA，双向寻呼机，手机或其他小型电子装置，它们将受益



于准确的高速文本输入。本系统可高效压缩用于键击序列多义性消除的大型数据库，并且在利用此经过压缩的数据库时不要求附加的处理带宽。本系统在应用于基于触摸屏的装置或具有有限数目的机械键并且也可能具有有限显示屏面积的装置时既高效又简便。

### 附图说明

本发明的上述各点及其他附加的优点在结合本说明书的附图参考下面的详细描述对其有所了解之后将很容易理解，附图中：

图 1 为采用本发明的简化键盘多义性消除系统便携式计算机的优选实施例的示意图；

图 2 为图 1 的简化键盘多义性消除系统的硬件方框图；

图 3A-3C 为用于简化键盘多义性消除系统的优选实施例多义性消除软件的流程图；

图 4 为本优选实施例用于判断哪一个文本对象应该用作供简化键盘多义性消除系统使用的树形数据结构的每个节点中的默认对象的软件流程图；

图 5A-5B 为示出随着每个键击执行的软件过程的优选实施例的流程图，该过程系用来对与简化键盘多义性消除系统的键击相关联的多个标点字符中间的所要求的标点字符进行多义性消除；示出在刚接收到键击时执行的处理及在键击的所有其他处理结束时执行的

处理；

图 6 比较了键与符号的物理相关性及包含出现在物理键上的字符的重音符变化的逻辑相关性的情况；

图 7 为联系逻辑符号与键索引的索引表示例；

图 8A 示出词汇模块的树节点中的数据内部优选配置法；

图 8B 示出指令的优选实施例的语义成分；

图 9 示出一个优选实施例中的节点结构中的可能内部数据项目的 4 个示例；

图 10 示出一个未压缩词汇模块的优选树形结构；

图 11 示出在从词汇模块内检索出来的过程中用于对象的中间存储的优选实施例的对象列表的状态示例；

图 12 示出的是用于在按键列表给定时从词汇模块检索文本对象的软件过程的优选实施例的流程图；

图 13 示出的是在单个按键给定时用于遍历词汇模块的树形结构及改变对象列表状态的软件过程的优选实施例的流程图；

图 14 为用于建立折叠压缩词汇模块的软件过程的优选实施例的流程图；

图 15 为折叠词汇模块的树形数据结构软件过程的优选实施例的流程图；

图 16 为用于定位词汇模块的树中与给定节点相比较具有最大冗余度的第二节点的软件过程的优选实施例的流程图；

图 17A-17B 为用于计算词汇模块的树中两个节点之间的冗余度的软件过程的优选实施例的流程图；

图 18A-18K 为在代表性应用中简化键盘多义性消除系统的优选实施例的示意图。

## 具体实施方式

### I. 系统构造及基本操作

图 1 示出的是应用于掌上型便携式计算机 52 中的根据本发明而形成的简化键盘多义性消除系统 50。掌上型便携式计算机 52 包含一个在触摸屏 53 上实现的简化键盘 54。对于这个应用，名词“键盘”的定义扩大为可包含具有键定义区的触摸屏，分立机械键及薄膜键等等的确定区域的任何输入装置。键盘 54 的数据输入键的数目比标准 QWERTY 键盘的少。在一个实施例中，键盘包含 16 个标准的全尺寸键，排列为 4 行 4 列。更具体而言，优选键盘包含 9 个数据键 56，排列成为 3×3 阵列，另外包含一个标点键 63，其中左手列是 3 个系统键 58，包括选择键 60，换档键 62 及空格键 64，以及位于最上一行 59 用来取消前一个键击的退格键 65。系统键的最上一行 59 还包

括 3 个标明 Accent (重音符), Numbers (数字) 及 Symbols (符号) 的模式键用来进入分别打入重音符、数字及符号的模式。

在图 1 中示出在键盘 54 中的每一个键上字母的优选排列法。图 1 也示出与用于英文的键盘 54 中的标点键 63 相关联的标点字符的优选排列法。

数据通过简化键盘 54 上的键击输入到多义性消除系统。当用户利用键盘输入键击序列时, 文本显示于计算机显示器 53。在显示器上定义两个区域显示给用户的信息。上边文本区 66 显示用户输入的文本并用作文本输入及编辑缓冲区。位于文本区下方的选择列表区 70 提供与用户输入的键击序列相对应的单词及解释的列表。正如下面将另外详细描述, 选择列表区 70 帮助用户解决输入键击的多义性。在另外一个优选实施例中, 此系统可应用于具有有限显示空间的装置, 并且在正在生成的文本中的插入点 88 只显示当前所选择的或最可能的单词对象。

图 2 示出简化键盘多义性消除系统的硬件方框图。简化键盘 54 及显示器 53 通过适当的接口电路与处理器 100 连接。在处理器上还可连接一个任选的扬声器 102。处理器 100 接受键盘输入并管理所有的到显示器及扬声器的输出。处理器 100 与存储器 104 连接。存储器是临时存储介质, 如 RAM, 软盘, 硬盘或 CD-ROM 的组合。存储器 104 中包含管理此系统运行的所有软件。最好是存储器包含操作系统 106, 多义性消除软件 108 及下面将要讨论的相关的词汇模块 110。还可以包括一个或多个任选应用软件 112, 114。应用软件的例子包含字处理软件, 软件字典, 以及外语翻译软件。语音合成软件也可作为软件提供, 以便使简化键盘多义性消除系统可用作通讯工具。

返回到图 1。简化键盘多义性消除系统 50 容许用户只利用一只手就可快速输入文本或其他数据。数据利用数据键 56 输入。每个数据键具有多个意义, 由键上的多个字母, 数字和其他的符号代表。(对于本公开申请, 每一个数据键由数据键的中央行的文本标志, 比如 DEF 标志右上数据键)。因为单个键具有多个意义, 键击序列的意义

是有多义性的。当用户输入时，各种键击解释就显示于显示器上的多个区域中以便帮助用户解决任何多义性问题。在具有足够的显示面积的系统，输入键击的可能解释的选择列表 76 在选择列表区 70 提供给用户。选择列表的第一条目 78 被选定为默认解释并显示于文本区 66 的插入点 88 处。在此优选实施例中，此条目在选择列表 76 及插入点 88 处均在周围加实线框显示。这种格式表示确定插入点对象和选择列表之间的视觉关系并且表示此对象是隐含选择的当前选择列表中最常出现的对象。另外一种替代方式是不提供选择列表，并且在插入点 88 处只显示默认对象（在激活“选择”键之前会第一个显示的对象），或当前选择的对象（如果显式地选定了的一个的话）。

输入键击的可能解释的选择列表 76 可以以多种方式排序。在正常运行模式下，在初始时将键击解释为字母条目来拼写单词（以下称之为“单词解释”）。因此，选择列表中的条目 78, 79, 80 是与输入的键击序列相对应的单词，其排序为与键击序列相对应的最常用的单词列在第一个。例如，如图 1 所示，由用户输入键击序列 ABC, GHI, 及 DEF。在键输入时，同时执行词汇模块查找以定位具有匹配键击序列的单词。从词汇模块认定的单词在选择列表 76 中显示给用户。单词按使用频率排序，最常用的单词首先列出。利用上述的示例键击序列，从词汇模块中认定单词“age”，“aid”及“bid”作为与键击序列对应的最可能的单词。在认定的 3 个单词中，“age”比“aid”或“bid”更为常用，所以列在首位。第一个单词也作为默认解释而暂时安置在插入点 88 处。在按下选择键 60 之前，取作默认解释的此第一单词采用同样的格式安置于插入点 88 及选择列表 76 中。比如，如图 1 所示，此单词在文本中以带有实线框的形式出现，实线框的大小为可以容纳该单词。在显示区不够大而难以在屏幕上列出实际的选择列表的系统中，潜在匹配词列表保存在存储器中，其排序系按照匹配文本对象的相对频率。

在此优选实施例中，在输入与所要求的单词相对应的键击序列之后，用户只要按下空格键 64 即可。默认单词（选择列表上的第一个

单词)立刻输出到插入点,而选择列表清空,并且一个空格也立即输出到文本区的插入点处。另外一种方式是任何其他可显式生成无多义性字符的(如进入符号模式并按下一个无多义性地与符号模式中的单个具体字符相关联的键)手段都可用来达到同一目的,例外是具体无多义性字符(不是空格键)是挂在插入点 88 处的输出单词之后。另外一种方式是可按下标点键 63 生成句点,短划或撇号,解释见下。

如选择列表中的第一条目不是键击序列的所要求的解释,用户可重复按下选择键 60 通过选择列表中的所有条目。按下选择键将重复显示选择列表 76 中的第一条目,其周围带有一个点线框,并且也在插入点 88 处重复显示第一条目,其周围的格式框完全相同。从意图上来讲,从实线框变为点线框表示通过用户按下选择键显式地进行的选中文本已经接近接受为正在生成的文本。如选择列表中的第一条目是键击序列所要求的解释,用户就继续利用数据键 56 继续输入下一个单词。如选择键还额外负担有空格生成功能,则在插入下一个单词之前生成一个空格。否则下一个单词的开始将与当前的单词末尾连接不插入空格。每按下一次选择键,选择列表中的下一个条目就加上点线框,并且该条目的复制条目将临时安置于插入点(代替原来临时安置的单词)并添加点线框。将下一个条目临时安置于文本区使得用户可以将注意力保持文本区而无须去参考选择列表。用户也可以选择使系统配置成为在接收到第一次按下选择键的信息时临时安置于插入点的单词可扩展(垂直或水平)为显示当前选择列表的复制列表。用户也可以选择在此复制的选择列表中显示最多的单词数。另外一种方式是用户可选择使选择列表一直在插入点处显示,甚至在第一次激活选择键之前。多义性消除系统将下一个单词的开始(由激活数据键 56 或生成显式无多义性字符发出信号)解释为肯定当前选择的条目为所要求的条目。于是所选定的单词就作为用户的选择保持于插入点,而其周围的框子彻底消失,并且该单词以正常字体显示不带特别格式。

如选择列表上的第二条目是所要求的单词,用户就可按下两次选

择键输入下一个单词，并且多义性消除系统自动将第二条目作为正常文本安置于文本区。如第二条目不是所要求的单词，用户可以检查选择列表并按照所要求的次数按下选择键来选择所要求的条目而输入下一个单词。当到达选择列表的末尾时，如继续按下选择键就会使选择列表滚动并在选择列表的尾部添加新的条目。在选择列表顶部的那些条目将从显示给用户的列表中消失。在用户按下任何一个数据键 56 来继续输入文本时经过多次按下选择键而选择的条目将自动地安置于文本区。另外一种方式是在输入与所要求的单词相对应的键击序列之后，用户可通过触摸选择列表从其中选择所要求的单词。所选定的单词立即输出到插入点不添加空格，并且选择列表清除。之后用户可按下空格键生成一个空格，该空格立即输出到文本区的插入点 88 处。

在大多数文本输入时用户的键击序列是要输入组成单词的字母。但是，应该了解，与每个键相关联的多个字符和符号可容许单个键击和键击序列具有数种解释。在优选简化键盘多义性消除系统中，在键击序列得到解释和作为单词列表显示给用户的同时，各种不同的解释将自动确定并显示给用户。

例如，键击序列可解释为与用户可能输入的可能的有效单词序列相对应的单词词干（以下称之为“词干解释”）。与单词解释不同，单词词干是不完整的单词。通过指明最后键击的可能解释，单词词干使用户可以很容易肯定输入的键击序列正确，否则，如果在单词输入过程中用户一时走神，可重新打入。如图 1 所示，键击序列 ABC GHI DEF 被解释为可组成有效词干“che”（可组成单词“check”，“cheer”等等）及“ahe”（可组成单词“ahead”，“ahem”等等）。于是在选择列表中提供词干解释作为条目 81 及 82。最好是词干解释按照通过添加数据键的附加键击每个词干可生成的单词集合的复合频率排序。待显示的此种条目的最大数目及最小复合频率可由用户选定或由系统配置，所以有些词干解释可能不显示。在当前的示例中，词干“bif”（可组成单词“bifocals”），“cid”（可组成单词

“cider”)未显示。当在选择列表中列出词干解释时,如词干解释与选择列表中的单词重复,则将该词干略去。但是,当词干被略去时,与略去的词干相对应的单词可以符号标志以便表示还存在以此单词作为词干的更长的单词。词干解释可通过肯定输入了可导致所要求的单词条目的正确键击而为用户提供反馈。

简化键盘多义性消除系统的运行由多义性消除软件 108 管理。在本系统的一个优选实施例中,选择键承担的“额外负担”是它既可以执行从选择列表中选择所要求的单词的功能,又可以在合适处生成显式空格字符。在此系统中,标志“OverloadSelect(额外负担选择)”(在步骤 164 及 174A 中援引)设定为“真”。在将单个的空格键定义为生成显式空格字符的系统中,标志“OverloadSelect(额外负担选择)”设定为“伪”。

图 3 为用来生成选择列表帮助用户对多义性键击序列进行多义性消除的多义性消除软件的主例程的流程图。在步骤 150,此系统等待接受键盘 54 的键击。在接受到键击时,在步骤 150A 系统执行在图 5A 中所示的标点键 63 所要求的预处理,这一点在下面详细讨论。在判定步骤 151 中,进行检验以判断所接受的键击是否是模式选择键。如是,则在步骤 172 中系统将标志设定为表示当前系统模式。在判定步骤 173 中,进行检验以判断系统模式是否改变。如是,则在步骤 171 中在需要时重绘键顶以反映当前系统模式。如系统模式未改变,或已改变并其后键顶经过重绘,则主例程返回步骤 150 并等待下一个键击。

另一方面,如步骤 151 判断键击不是模式选择键,则在判定步骤 152 中,进行检验以判断所接受的键击是否是选择键。如不是选择键,则在判定步骤 153 中,进行检验以判断系统是否是处于特殊显式字符模式,如显式数字模式。如是,则在步骤 167 中,接受该项目并作为正常文本输出。如临时接受的项目不在选择列表中,或如是并且已经接受,则在步骤 168 中,与键击相对应的显示字符输出到文本区。其次,在判定步骤 169,进行检验以判断系统模式是否自动改

变，就像在符号模式的场合那样。如是，执行转到步骤 170 和 171，其中系统模式返回到先前是活动的模式并且键顶进行相应的重绘。之后执行返回步骤 150。

如在步骤 153 中无显式字符模式是活动的，则在步骤 154 中，此键击添加到存储的键击序列中。在步骤 156 中，与键击序列相对应的对象从本系统中的词汇模块中认定。词汇模块是与键击序列相关联的对象的库。一个对象是根据检索的键击序列要检索的任何一片存储数据。例如，在词汇模块内的对象可包含数字，字母，词干，词组，或系统功能和宏。这些对象的每一个简要地描述在下表中：

对象	相应的数据
数字	每一位相应于单个键击的数字，比如两位序列“42”。
字母	与成对键击相对应的字母与字母序列，比如 3 字母序列“str”。各对键击用于利用输入单个字母的两键击指定方法来进行多义性消除。
单词	与单个或多个键击相对应的单词，比如 4 字母单词“done”
词干	代表组成一个单词的更长的字母序列的有效部分的字母序列，比如“able”作为单词“albeit”。
词组	用户定义的或系统定义的与单个或多个键击相对应的词组，比如“To Whom it May Concern:”。



系统宏	描述系统或用户定义的功能的单词和相关的代码，比如用来清除当前文本区的“<clear>”。除了描述词，在词汇模块中系统宏对象与执行指定功能所必需的执行代码相关。
-----	---

上面讨论了优选词汇对象，可以理解，还可以考虑其他的对象。例如，图像对象可与所存储的图形图像相关联，或语音对象可与所存储的语音段相关联。还可以预见拼写对象为可以将通常误拼的单词的键击序列及打字错误与单词的正确拼写联系起来。例如，包含字母序列“ie”或“ei”的单词将出现在单词列表中，即或这些字母的键击意外地会与其正确序列相反。为了简化处理，每个词汇模块最好是包含类似的对象。但可以理解，各种对象可在词汇模块内混合。

在图 6 中示出单个键 540 的代表性示意图。在优选实施例中的键的内部逻辑表示不需要与物理配置完全相同。比如逻辑键 541 是与法文词汇模块相关联的键的优选逻辑描述。法文字母表需要三个重音符 542 (ÂÇÀ)。另外，这些符号 543 最好是按照其在法文词汇中的使用频率降序索引。扩展时，图 7 是联系逻辑符号与用于对输入法文单词的按键进行多义性消除时所使用的键索引的优选表。

在图 10 中示出单词对象词汇模块 110 的代表性示意图。根据相应的键击序列使用树形数据结构来组织词汇模块中的对象。如图 10 所示，词汇模块中的每个节点 N1, N2, . . . N9 代表一个具体键击序列。树中的节点通过路径 P1, P2, . . . P9 连接。由于在此多义性消除系统实施例中有 9 个多义性数据键，词汇模块树中的每个父节点可与 9 个子节点相连接。通过路径连接的节点代表有效键击序列，而节点缺乏路径代表无效键击序列，即不与任何存储单词相对

应。

词汇模块树根据接收到的键击序列进行遍历。比如，按下自根节点 111 的第一数据键使与第一数据键相关联的数据从根节点 111 内部取出并进行评估，之后遍历通到节点 N1 的路径 P1。在按下第一数据键之后按下的九数据键使与第九数据键相关联的数据从根节点 N1 内部取出并进行评估，之后遍历通到节点 N19 的路径 P19。正如下面将详细描述，每个节点与多个与键击序列相对应的对象相关联。当接收到每个键击并处理相应的节点时，由与键击序列相关联的对象生成一个对象列表。多义性消除系统的主例程利用每个词汇模块的对象列表生成选择列表 76。

图 8A 是与每个节点相关联的节点数据结构 400 的框图。数据结构包含连接词汇模块树中的每个父节点到子节点的信息。此数据结构还包含用于认定与节点所代表的具体键击序列相关联的对象的信息(指令)。

在节点数据结构 400 中的第一字段是有效键位字段 402，它表明子节点的数目与标志，子节点与父节点连接以及 9 个可能键中的哪些与认定与节点所代表的具体键击序列相关联的对象的信息(指令)相关联。因为在此优选实施例中有 9 个数据键，所以最多可以有 9 个子节点与任何一个父节点连接，从而在有效键位字段中提供 9 个有效键位来表示存在或不存在子节点。每个有效键位与包含词汇模块中的相应各子节点数据结构的指针的指针字段 404a, 404b, . . . 404n 相关联。因为子节点只在与子节点相关联的键击是与父节点相关联的键击序列的有效继续才存在，所以指针字段的数目对每个节点是不同的。比如，有效键位字段 402 可表示 9 个可能键击只有 6 个导致有效子节点。因为只有 6 个有效路径，在父节点的数据结构中只包含 6 个指针字段。有效键位字段 402 是用来查明包含在节点数据结构内的指针字段的标志。如键击不导致有效子节点，则相关联的指针字段从节点数据结构中去掉以便节省用来存储词汇模块所需要的存储空间。

与每个节点相关联的有多个与节点所代表的键击序列相对应的对象。每个对象由附属于具体有效键的包(分组)408中的指令406描述,该有效键由包含在节点数据结构中的有效键位字段402中的位模式指明。

每个包408中的每个指令406描述与每个节点代表的键击序列相对应的对象中的一个。描述一个对象要求维护两个对象列表。图11示出由多义性消除软件过程由词汇模块树中的父与子动态生成的代表性对象列表。对象列表430是一个包含与代表2个键击的节点相关联的对象 $1-N_1$ 的对象列表。对象列表440是一个包含与代表3个键击的节点相关联的对象 $1-N_2$ 的对象列表。每个对象列表包含与每个节点相关联的所有对象的列表。对象列表430与代表图1的键盘的键击序列ABC ABC的父节点相关联。对象列表440与代表键击序列ABC ABC TUV的子节点相关联。可以理解,对象列表的大小是根据与每个节点相关联的大小的实际数目而变化。

与子节点相关联的每个对象是通过在为父节点构造的对象上添加一个字符序列而构造的。因此,图8A中的指令包408就包含一个带有示于图8B的OBJECT-LIST-INDEX字段556的指令558,可用来从父节点对象列表中认定用于构造子节点对象的对象。比如,参考图11,在老对象列表430中的第一对象“ba”用来构造新对象列表440中的第二对象“bat”。原来的对象标志符字段OBJECT-LIST-INDEX 556就可提供到老对象列表中的条目的链接以认定用来构造新对象的老对象。

指令558也包含LOGICAL-SYMBOL-INDEX字段555以指示要添加到认定对象的符号以便构造新对象。因此LOGICAL-SYMBOL-INDEX字段就从节点键击序列中的最后键指定将要添加以构造新对象的字母。此字母是由图7所示的表指定,其中的LOGICAL-SYMBOL-INDEX字段对应于表550的第一行中的逻辑符号索引552,并且指定键出现的行是由此表的第一列中的给定的键索引认定。比如,参考图11,在新对象列表440中的第一对象“CAT”是利用老对象列表430中的

第二对象“CA”和添加一个指定 T 的附加键击而构造。在图 7 的逻辑符号索引表中“T”是在 TUV 键上的第一逻辑字母，因此生成对象“CAT”的指令的 LOGICAL-SYMBOL-INDEX 字段设定为 1 以指示此表中的第一字母。以这种方式对对象进行编码是利用与每个节点相关联的已知键击序列及字母到键的已知关联以便大大减少每个词汇模块所需的存储空间。

这一词汇编码技术也容许不经检索而访问词汇模块条目。在接收到每个新有效键击时系统执行与当前节点上的键相关联的指令以便从老对象构造新对象，之后按照单个指针到达适当的子节点。另外，不需要将每个对象存储于词汇模块中，新对象是利用 LOGICAL-SYMBOL-INDEX 字段定义而添加到老的解释上。这样，由词汇模块中多个对象共享的单词词干就只存储一次而能够原来生成所有由它派生的对象。公开的这种存储方法要求维护词汇模块树中的父节点对象列表以便构造子节点的对象列表。

如图 7 所示的逻辑符号索引表中的条目不一定必须是单个字符-任意序列可占据一个单个条目。比如，ASCII 字符串“tlike”可添加到老对象列表的第二对象“ca”而形成“catlike”。采用这种方式，输入键击序列的长度不一定必须与相关联的对象的长度直接对应。存储于符号索引表内的条目中的 ASCII 序列将容许词汇对象由任意的键序列认定，即存储于词汇模块树内的任意位置。

可以将具有任意键击序列的对象存储的能力用来加速系统对缩写和缩略字的处理。缩写和缩略字可由与其忽略标点的纯字母内容相对应的键击序列认定。结果用户可以不必输入标点而很容易访问缩写和缩略字，这可显著地节约键。比如，用户输入“did' t”的键击序列时可不必输入“n”和“t”中间的撇号。词汇模块中与键击序列“didnt”相对应的指令援引在“n”和“t”中间有撇号的 ASCII 序列作为表中的唯一一个符号。所以此多义性消除系统将自动向用户显示正确的单词“did' t”而无需用户输入标点字符。此多义性消除系统利用同一个表可恰当地显示具有独特字符的外来词(如“Ü”

可以作为 TUV 键上的单个键击输入)。大写可以以类似方式处理。字母永远大写的单词，首字母大写的单词，单词中间字母大写的单词可与去掉表示大写的键击序列相关联，用户就无需输入此类大写。在每个指令 558 中还可包含对象类型字段原来指定关于正在构造的对象的附加信息。此对象类型字段可包含一个指定所生成的对象是单词，单词词干或其他对象的代码。所以词对象类型字段容许将不同类型对象在给定的词汇模块中混合。此外，对象类型字段也可包含有关词类的信息，有关对象如何大写的信息或构造各种词形变化和词尾所需的信息。利用具有词类信息的词汇模块的简化键盘多义性消除系统可利用附加信息进行句法分析来改进多义性消除过程。对象类型字段也可包含独特代码以容许传输压缩形式的文本。此独特代码可传输到远方终端代替传输输入的键击序列或相关联的多义性消除字符。

此优选词汇模块树形数据结构的关键特点之一是与每个节点相关联的对象是存储于节点数据结构 400 中按其使用频率排列。即由包 406 中的第一指令构造的对象具有比由包 406 中的第二指令(如存在)构造的对象高的使用频率，而后者又具有比由包 406 中的第三指令(如存在)构造的对象高的使用频率。这样，对象在对象列表中自动按使用频率降序排列。此描述中所谓单词对象使用频率指的是在具有代表性的应用语料库内使用给定单词的概率，该值与每个单词在语料库中出现的次数成正比。在单词词干对象的场合，使用频率由共享该相同词干的所有单词的频率的总和确定。

将使用频率或其他等级信息存储于每个节点可避免在使用系统时需要判断和按每个对象的等级排序。这一点在单词对象词汇中有重要意义，因为存储的对象可包含大量更长的单词共有的共享词干。动态地判断这些词干的相对等级(rank)将要求子节点的整个树并累积关于每个词干的信息，为便携式计算装置添加明显的处理开销。此外，当使用频率或等级由节点 400 中的对象的排序隐含表示时，这一信息不需要另外的存储空间。

虽然对象在节点数据结构 400 内的存储是按照使用频率排序,但可以理解使用频率字段也可以与每个指令相关联。使用频率字段将包含与此相关对象的使用频率相对应的代表数。不同对象之间的使用频率将通过比较每个对象的使用频率而确定。使用将使用频率字段与每个对象包相关联的这后一种构造方法的优点是使用频率字段可由多义性消除系统改变。比如,此系统可改变使用频率字段以反映用户在代表性文本输入期间用户使用词汇模块内的某些对象的频率。

返回到图 3,在标志 156 中,与接收到的键击序列相对应的那些对象在每个词汇模块中认定。图 12 示出的是用于分析所接收的键击序列以便认定在具体词汇模块中的相应对象的子例程 600 的流程图。子例程 600 构造具体键击序列的对象列表。步骤 602 清除新的对象列表。步骤 604 在其根节点 111 启动遍历树 110。步骤 606 取得第一键的按下。步骤 608 到 612 形成处理所有的可获得的关键的按下的循环。步骤 608 调用图 13 的子例程 620。判定步骤 610 判断是否所有的可获得的关键的按下都已经处理。如果有任何键的按下尚未处理,步骤 612 转到下一个键的按下。如所有的键的按下都已经处理,步骤 614 返回到完成的对象列表。可以理解,如主例程在处理每个新键击序列时都重复调用子例程 600,每个新键击序列比最后一个多一个键并且除最后一个以外的所有的键都与上一次启用相同,则如果子例程 620 直接被调用来只处理最新一次键按下,初始化步骤 602,604 可以绕过。

图 13 示出的是从子例程 600 调用的子例程 620 的流程图。在图 3 所示的主例程中键击由系统在步骤 150 中检测。如果存在通到与键击相对应的子节点的有效路径,则接收到新键击就会引起在词汇模块树中的下行遍历。在图 13 中的步骤 621,就检查节点数据结构 400 的有效键位字段以判断是否有效指令和指针与所接收的键击对应。在判定步骤 622,检查有效键位字段以判断是否存在由指令 406 和指针字段,如 404a,组成的有效包 408 与输入的键击相对应。如果没

有有效包与键击对应，在步骤 624 老对象列表将返回到主例程以生成选择列表，因为接收到的键击是不与词汇模块内的任何对象对应的无效键击序列的一部分。由步骤 622 及 624 构成的主例程 620 的分支就忽略任何无效键击序列并返回在父节点生成的对象列表以便在多义性消除系统生成的选择列表中进行可能的插入。

如在步骤 622 存在与所接收到的键击相对应的有效包，子例程转到步骤 626，在该步骤新对象列表复制到老对象列表。如上所述，为了构造新对象列表，多义性消除系统从复制老对象列表开始。因此在步骤 626，将来自前一个节点的对象列表存储以便可原来构造新对象列表。

步骤 628 取出与给定键相关联的第一有效指令。步骤 630 将迭代器 NEW-INDEX 初始化为 1 以使第一指令将在新对象列表中生成第一条目。之后子例程进入包括步骤 632 至 642 的循环以构造与有效指令相关联的对象列表。在步骤 632 中，检查 OBJECT-LIST-INDEX 字段 556 并从老对象列表装入相应的对象。在步骤 634 中，检查 LOGICAL-SYMBOL-INDEX 字段 555 并将合适的符号(通过像 550 这样的图 7 所示的逻辑符号索引与接收的键相关联)添加在认定对象的末端。可以理解，在步骤 634，如果在给定键 551 向符号表 550 和逻辑符号索引 552 的输入保持一个字符序列，则长度大于 1 的 ASCII 序列可添加到认定的对象上。在步骤 636，组合对象和符号作为新对象存储于新对象列表。在步骤 638 中进行检查以判断子例程是否已经处理在给定节点与给定键相关联的最后有效指令。如最后有效指令尚未处理，则在步骤 640 中取出下一个指令。在步骤 642 中，NEW-INDEX 加 1。

如在判定步骤 638 中的检查表明对此节点所有的对象都已经构造，则子例程转向步骤 644 并按照相关联的指针到子节点。在步骤 646 新对象列表返回到主例程以生成选择列表。可以理解，用来生成与每个节点相关联的对象列表的主例程 600 对于从用户接收到的每个键击都执行一次。在用户输入新键击序列时，不进行词汇模块的

“检索”，因为每个键击只是使子例程在词汇模块树内前进一个层次。因为不对每个键击序列进行检索，词汇模块以最少的处理开销将与每个节点相关联的对象的列表返回。

可以理解，在词汇模块对象及键击序列之间的这种关系是词汇模块的使用细节。当处理与当前输入键序列相关联的节点时，可能要遍历附加的子节点以便认定具有以输入键击序列开始的键击序列并且其出现相对频率大于某一阈值的候补对象。此阈值可根据当前节点的特性动态地进行调整，比如根据其是否生成足够的对象来填充显示器上的选择列表区 70。对象的认定是通过在词汇模块树中沿有效路径遍历一直到对象认定为止。此种候补对象可在由输入单词列表构造数据库时认定，并且与候补单词完成相对应的节点和指令可进行标记使系统能够识别何时对该节点的遍历对应于候补单词的生成。此专门标记是需要的，因为同一节点和指令可能在下述的压缩过程中在树形结构转换后在处理对应于不同单词的不同的输入键击序列时受到遍历。为区分候补单词序列与其他序列只需要增加足够的信息，如序列长度或在此键序列中的已知位置的特殊键值。另外一种替代办法是候补单词的端节点可进行特别标记使其不会与用于其他单词的节点合并，见下述。于是多义性消除软件可在当前节点的后代的限定的附近内向前搜索这些候补单词，如发现，就可在所有的相应于对象的键击输入之前将其置于选择列表中。这些对象是在与输入的键击序列直接相关联的对象之外包括进来的。在选择列表中显示与较长键击序列相关联的对象(称之为“前视”特征)使得用户可以有选择地立即选定对象而无需结束剩余的键击来指定该对象。

返回到图 3。在步骤 158-165 中，通过检查词汇模块中的键击序列而发现的对象被授予优先级并在选择列表 76 中显示给用户。为了判断在选择列表中显示的对象次序，在每个词汇模块之间以及在从每个词汇模块返回的列表之间建立优先级。

为了给从各种词汇模块认定的对象列表确定优先级，在步骤 158



中检查简化键盘多义性消除系统的运行模式。如上所述，在正常运行模式中，单词解释首先显示于选择列表中。来自单词词汇模块的对象列表将被赋予比来自其他词汇模块的对象列表更高的优先级。反之，如多义性消除系统是在数字运行模式下，数字解释将被赋予比其他词汇模块更高的优先级。因此，多义性消除系统的模式确定词汇模块对象列表之间的优先级。可以理解，在某些模式中，词汇模块对象列表可整个从选择列表中去掉。

由词汇模块生成的对象列表可能只包含单个一个条目，或是包含多个条目。在步骤 160 中，如对象列表包含多个条目，就决定来自同一词汇模块的对象之间的优先级。对于与在给定的词汇模块中查寻的具体键击序列匹配的对象也赋予优先级，该优先级确定其互相间的相对显示。如上所述，最好默认显示顺序是由具有使用代表性的语料库中的使用频率的降序。与每个对象相关联的优先级数据可用于将对象在选择列表中排序。因为选择列表区 70 可显示的条目数目有限，在低于预定最小使用频率的对象可能不在选择列表的初始显示中。这些未显示的条目以后会在用户滚动到显示列表尾部之外时添加到此选择列表中。选择列表自动滚动以使当前选择的对象一直可见。用户也可以利用专用滚动按钮进行手工滚动而使另外的对象进入视图，在此场合当前选择的对象可能从视图中滚出。另外一种替代的方式是选择列表上的所有的对象可按照用户的要求同时显示在“下拉式”列表中。

与在词汇模块中查寻的对象的显示相关联的很多属性是可以通过访问适当的系统菜单而由用户设定的。比如，用户可指定在选择列表区中各个对象的顺序或对象的类别。用户也可以设定判定词汇模块之间和从每个词汇模块中认定的对象之间的优先级。这样，在选择列表区中显示给用户的条目的数目可保持最少。在选择列表区中的附加条目永远可通重复按下选择键滚动进入视图。

在决定了对象之间的优先级之后，在步骤 165 中由认定的对象构造选择列表并显示给用户。作为由用户输入的多义性键击序列的默

认解释，选择列表中的第一个条目临时突显置于文本区 66 的插入点 88 处。之后多义性消除软件例程返回到步骤 150 中等待下一个键击。

返回到步骤 152。如检测到的键击是选择键 60，则从判定步骤 150 选取“yes(是)”分支转向判定步骤 163 检查判断当前的选择列表是否为空。如是，则在步骤 164，如 OverloadSelect 标志设定为“真”时，就生成一个显式空格并立即示出到文本区。OverloadSelect 标志是一个系统标志，在不包含显式空格键 64 的系统中此标志设定为“真”，并且在此系统对任何激活选择键连续序列的第一个激活生成一个空格字符，或是如果在选择键激活时选择列表为空时也生成一个空格字符。在对接收到的键击进行这种初始处理之后，在步骤 164B 系统执行图 5B 所示的对标点键 63 所要求的后处理，详情见后。之后执行返回步骤 150。如在判定步骤 163 中，选择不为空，则选取“no(否)”分支而转向步骤 174A。在步骤 174A 中，如 OverloadSelect 标志设定为“真”，在选择列表的每个文本项目的尾部和在插入点添加一个空格。在步骤 174 中，围绕选择列表中的第一条目(以及在临时安置该条目的插入点)的实线框改变为点线框。在图 3 所示的步骤 175 中，系统等待检测用户输入的下一个键击。在接收到键击时，在步骤 175A 系统执行在图 5A 中所示的对标点键 63 所要求的预处理，详细讨论见后。在判定步骤 176 中进行检查以判断下一个键击是否是选择键。如下一个键击是选择键，则在步骤 178 中系统前进到选择列表中的下一个条目并将其标记为当前选定项目。在对接收到的键击进行这一初始处理之后，在步骤 178B 中系统执行图 5B 所示的对标点键 63 所要求的后处理，详情见后。在步骤 179 中，当前选择的条目临时显示于选择列表中和在插入点处，其周围带有一个点线框。之后例程返回到步骤 175 检测用户输入的下一个键击。可以理解，由步骤 175-179 形成的循环容许用户通过多次按下选择键选择所输入的具有较低的使用频率的多义性键击序列的各种解释。

如下一个键击标志选择键，则例程从判定步骤 176 继续转向步骤

180, 在该步骤中将临时显式的条目选择为键击序列的解释并在文本区中转换为正常文本格式。在步骤 184 中, 从存储器中清除老的键击序列, 因为在选择键之后接收到多义性键击向系统表示一个新的键击序列的开始。之后在步骤 154 中利用新接收到的键击开始新的键击序列。因为具有较高使用频率的单词解释是作为默认选择显示, 多义性消除软件的主例程使用户可继续输入文本, 而只有在很少的场合需要另外激活选择键。

如上所述, 在正常运行模式下选择列表 76 中对应于单词的条目首先列出。在其他情况下, 有可能希望其他的键击序列解释在列表中首先列出。比如, 在显示区不足以支持在显示屏上显示文本选择“列表”的系统中, 取自数据库的可能匹配对象的集合由软件存储于内部并且在插入点只显示默认对象或显示选定对象(在一次或多次激活选择键之后)。

在此种系统中, 在插入点显示的对象可能随着在输入序列添加的每一个键击发生显著的改变, 因为与新的键击序列相对应的使用频率最高的单词可能与前一次输入的序列所对应的最常用的单词外形很少或毫不相象。结果可能是在插入点处显示的对象对用户可能表现为“跳动”, 并且其不断改变的外表可能令人分心, 尤其是对于新用户或不熟悉的用户。为了减轻在插入点的这种“跳动”, 系统可配置成为不是永远显示与当前键序列相对应的频率最高的单词对象, 而可以显示一个单词或单词词干, 如果它对应于一个较长的单词或总频率超过阈值的多个单词。这种“词干提升”的结果是在插入点的显示的单词的改变经常仅仅是在原来显示的文本对象上添加一个字母。比如, 在打入“this”一词时用户使用 4 次键击 TUV, GHI, GHI, PQRS。不执行“词干提升”时, 系统将会在插入点显示如下的对象序列(每次一个): “t”, “vi”, “ugh”, “this”。执行“词干提升”时, 系统将会在插入点显示如下的对象序列(每次一个): “t”, “th”, “thi”, “this”。结果是新用户或不熟悉的用户将会更相信多义性消除系统将会正确地解释他或她的键击。

图 4 的流程图示出系统如何判断对给定的键击序列在给定节点显示哪一个对象。

图 4 为示出在每个节点对应于一个多义性键击序列的树形结构的节点中生成单词数据库的的软件程序例程的流程图。在每个节点，生成多个可与由键击序列生成的可能的单词或单词词干相对应的单词对象。在步骤 190 中，设定阈值“MinFactor”。此阈值对应于一个最小因子，此因子规定文本对象要想提升超过单词对象（即成为在该节点的选择列表中首先显示的默认对象）其总频率必须超过在节点处频率最高的单词对象的总频率的倍数。单词对象的总频率是单词本身的频率加上以该单词为词干（即对应于较长单词的几个起始字母）的所有更长的单词的频率的总和。在步骤 191 中，设定第二阈值“MaxWordFreq”，该阈值对应于一个单词的最大相对频率，另外一个词干的值超过此值就可以提升到该单词之上。在步骤 192 中，树中的所有节点受到遍历并且标记为未处理。在步骤 193 中，例程判断是否树的所有节点都已经处理过。如是，则例程停止，否则在步骤 194 中，系统取出下一个未处理节点并判断频率最高的单词对象的频率（MostFreqWord），该单词及以该单词为词干的所有单词的频率的总和（MostFreqWordTot），以及以某一对象为词干的所有的单词的总频率在与节点相关联的所有对象中最高的该对象（MostFreqStemTot）。在步骤 195 中，系统检查是否 MostFreqStemTot 至少超过 MostFreqWordTot 值 MinFactor 倍。如否，与节点相关联的默认对象就不改变，而系统返回步骤 198，在该步骤节点标记为已经处理之后返回步骤 193。如在步骤 195 中 MostFreqWordTot 被超过至少 MinFactor 倍，系统就检查是否与节点相关联的当前默认对象的频率（MostFreqWord）超过另外一个对象可以提升到一个单词之上的最大值（MaxWordFreq）。如是，与节点相关联的当前默认对象就不改变，并且系统返回步骤 198。如否，在步骤 197 中节点改变，指定与 MostFreqStemTot 相关联的对象为与节点相关联的默认对象，之后返回步骤 198。

需要步骤 196 中的检查来防止很常用的词干提升到常用单词之上。比如, 单词“for”的词干“fo”具有很高的频率, 因此对于合理的 MinFactor 值(比如 1.5), 在步骤 195 中的检查定会满足, 即使常用词“do”也在同一节点中出现。单词“do”明显具有相对高的频率, 并且如果“do”不作为该节点的默认对象出现, 则对多义性消除系统的效率将是巨大的伤害。

## II. 先进的系统特征

### 1. 标点的多义性消除

图 1 示出本发明的一个优选实施例, 其中数据键 56 的左上键(“标点键”)与标点字符句点, 短划和撇号相关联。所有这些字符在多种语言中很常用, 比如在英文和法文中。其他字符集合(比如字符逗号, 短划即破折号)可赋予此键并利用与本发明的多义性消除系统方法同样的原理消除多义性。此系统利用围绕标点键 63 键击的键击的上下文来判断应该是使用哪一个标点。标点键 63 默认行为是可以修改以适应处理某些语言的某些特例。

图 5A 和图 5B 的流程图示出系统在激活标点键时如何判断生成哪一个字符。利用与标点键 63 相关联的不同的标点字符生成的不同文本对象添加到选择列表中。激活标点键 63 的默认解释由哪一个对象作为第一对象添加到选择列表来确定。也可以通过激活选择键或通过显示的选择列表中选择所要求的解释来选择另外的替代解释。图 5A 示出在示于图 3 的正常系统处理(其中系统对当前输入的每个键击序列添加应该键击)之前处理系统接收到的每个键击的例程判断哪些单词对应于新的输入序列并根据此新序列生成更新的选择列表。在图 5A 中示出的处理出现在图 3 的步骤 150A 和步骤 175A 中。图 5B 示出对新生成的选择列表进行后处理的例程。在图 5B 中示出的处理出现在图 3 的步骤 164B 和步骤 178B 中。假设所有的标志在系统初始化时都已清除。

在开始对每个键击进行预处理时, 在图 5A 的步骤 200 中, 检查

标志 ClearPunctMode。设定标志 ClearPunctMode(在步骤 262)以表示在处理下一个键击之前所有的标点检相关标志击缓冲区应清除。如在步骤 200 发现此标志设定,则在步骤 202 将专门用于标点键 63 处理的所有的标志及缓冲区清除。同样的标志击缓冲区在系统初始化时也清除。curPunctWorld 缓冲区是用来存储标点键软件将添加标点字符以生成添加到选择列表的文本对象。pState 标志存储标点键软件的当前状态并区分标点键 63 激活的上下文的不同的可能场合。如在步骤 204 中正在处理的当前键是标点键 63,则在步骤 206 当前默认(或显示选择对象)存储在 curPunctWorld 缓冲区,并且在步骤 208 中将标志 pState 设定为 PUNCT\_KEY,表示标点键 63 是最后接收到的键。步骤 212 代表如图 3 所示的从步骤 151 或 176 开始的键击的正常处理。之后从图 5B 所示的连接器 D 开始继续执行后处理。

如在步骤 204 中当前键不是标点键,则在步骤 220 中,如 pState 未设定,则在步骤 222 中将专门用于标点键处理的所有的标志及缓冲区清除,并在步骤 212 中继续键击的正常处理。如在步骤 220 中 pState 标志设定为非零值,则在步骤 230 中如当前键是多义性数据键 56 中的一个,则系统进到步骤 232 检查系统 Apostrophe\_S 是否设定。只有对像英文这样的语言 Apostrophe\_S 标志设定为“真”,在英文中通常在词尾在撇号后跟有字母“s”(形成所有格或单词“is”的短缩形式)。如 Apostrophe\_S 标志设定为“真”,则如在步骤 234 中发现 pState 标志设定为 PUNCT\_KEY(表示前一个键的激活所标点键)并且如在步骤 236 中当前键是与字母“s”相关联的数据键(图 1 中的 PQRS 键),则在步骤 238 中将 pState 设定为 APOS\_S,表示接收到标点键 63 并识别到在撇号后面跟有“s”的特别情况。如任何一个检查(步骤 234 和步或步骤 236)都失败,则在步骤 237 中将 pState 标志设定为 PUNCT\_RCVD,表示接收到标点键 63 并且将执行标点键处理,但是当前无特别情况(如识别到在撇号后面跟有“s”)。在任何一种场合,在 pState 设定之后,都将在步骤 212 中继续进行键击的正常处理。

如在步骤 230 中，判断当前键不是多义性数据键，则在步骤 250 中系统判断当前键是否是退格(BackSpace)键。如是，则在步骤 252 中将标点键相关标志及缓冲区恢复到其在被删除的键击之前的状态，之后在步骤 212 中继续(经过连接器 C)进行键击(退格键)的正常处理。如在步骤 250 中当前键不是退格键，则在步骤 260 中，执行一次检查判断当前键是否是 MODE(模式)键，选择键或换档键(即不会导致当前单词被“接受”和输出到文本缓冲区)。如是，在步骤 212 继续(经过连接器 C)进行键击处理。否则，在步骤 260 中判断此键是一个将导致此当前单词被接受的键，并且在步骤 262 中，清除 pState 标志和将 ClearPunctMode 设定以使所有的标点键相关标志及缓冲区在接收到下一个键击将在标志 202 中清除。

如在步骤 232 中判断 Apostrophe-S 标志未设定，则在步骤 270 中系统判断 Apostrophe-Term 是否设定，并在步骤 272 中判断 pState 是否设定为 PUNCT-APOS，表示标点键 63 接收到且识别到以撇号结尾的单词已经添加到选择列表的特别情况。Apostrophe-Term 标志设定为“真”是对像法文这样的语言，在法文中通常将各种一撇号结尾的单词(例如 l'，d' 等等)与后面的单词连接而不在两个单词中间添加空格。在这种语言中，使系统自动生成隐式“Select”以使用户能够立即开始打入后面一个单词而无需显式激活选择键是很方便的。如则在步骤 270 和 272 中的检查满足，则接受当前默认或选择的单词并输出到文本缓冲区就像激活了选择键一样。之后在步骤 276 中，清除当前键击列表，执行转向步骤 262，在其中清除 pState 标志并设定 ClearPunctMode 标志。另一方面，如在步骤 270 中系统判断 Apostrophe-Term 未设定，则在步骤 272 中 pState 设定为 PUNCT-RCVD，并且在步骤 212 中继续进行键击的正常处理(经过连接器 C)。

在图 5A(详细情况示于图 3)中的步骤 212 表示的键击正常处理之后，从连接器 D 开始执行如图 5B 所示的后处理。在步骤 300 中，如 pState 标志未设定，则不要求后处理，于是子例程就返回步骤 320。

步骤 302, 304 和 306 检查 Apostrophe-Term 是否设定, pState 标志是否设定为 PUNCT-KEY 和当前默认单词是否是以撇号结尾。如 3 个条件都满足, 则在步骤 308 将 pState 标志设定为 PUNCT-APOS, 否则系统转到步骤 310 而不设定新的 pState 值。如在步骤 310 中, pState 标志设定为 PUNCT-APOS 或 PUNCT-KEY, 则在步骤 312, 314 和 316 中将存储于 curPunctWorld 缓冲区中的文本对象在分别后接句点, 短划和撇号之后添加到选择列表。如在步骤 310 的检查失败, 并且如在步骤 330 发现 pState 设定为 APOS-S, 则在步骤 332 中将存储于 curPunctWorld 缓冲区中的文本对象在后接撇号及 's' 之后添加到选择列表。之后在步骤 334, 336, 338 和 340 中将存储于 curPunctWorld 缓冲区中的文本对象在每个与 PQES 键相关联的字母的后面后接撇号之后添加到选择列表。之后在步骤 342 中将 pState 标志设定为 PUNCT-RCVD。

如在步骤 330 中 pState 标志未设定为 APOS-S, 则在步骤 352 中系统检查当前选择列表是否为空。如否, 则是在数据库中发现一个或多个对象与键击缓冲区中的当前键序列(至少包含一次标点键激活)显式匹配。在此场合, 子例程就返回到步骤 364 中而不改变键击缓冲区中的内容以使数据库对象可拼写为好象那就是用户的意。否则, 只要在数据库中不再存在匹配对象, 就在步骤 356 中将存储于 curPunctWorld 缓冲区中的文本对象在后接短划之后输出到选择列表。在步骤 358 中, 将从键击缓冲区的开始直到并包含标点键的键击为止的键击删除。之后在步骤 360 中根据修改过的键击缓冲区查询数据库并生成更新的选择列表。最后在步骤 362 中在返回到子例程之前将标点键的所有的缓冲区和标志清除。

## 2. 在词汇模块存储器最小化的同时使性能最大化

示于图 13 的以示于图 10 的树形数据结构运行的子例程 620 的对象认定软件过程的键击组合中固有的几种新手段可用来检索更多的



对象词汇而同时又可利用较少的词汇模块存储器，并且不会增加子例程 620 的处理时间。

根据输入词典中符号的使用频率将给定词汇模块的逻辑符号索引表 550 的每一行中的符号进行排序，可以使树形数据结构 110 中的所有节点 400 的指令 558 的大多数具有等于 1 的 LOGICAL-SYMBOL-INDEX 字段 555。与此类似，通过将所有节点 400 中的像 406 这样的所有的包的指令 558 进行排序而使词干及单词对象在对象列表 440 中以其在该语言中的使用频率的降序生成，可以使树形数据结构 110 中的所有节点 400 的指令 558 的大多数具有等于 1 的 OBJECT-LIST-INDEX 字段 556。这样，在树 110 中的很多数据是冗余的。系统地认定冗余度和通过将链接父节点与子节点的路径重新指向并删除不再被引用的子节点消除冗余度可导致比原始树包含少得多的节点，少得多的指令和少得多的链接而仍然可从原始树检索到可检索的每个对象的折叠或卷绕数据结构。另外，经过其指令在对象列表 440 中生成类似对象的原始树的清楚的路径实例归并成为折叠树的公共路径，此后它们将作为通用(与专用的相对)的对象构造规则，使简化结构可以比原来用于定义给定词汇模块的树 110 的结构生成多得多的对象。比如，在一个优选实施例中由具有 30000 英文单词的列表生成的非折叠词汇树可能包含 78000 条指令。经过折叠过程的优选实施例的折叠之后，如多义性键击序列和检索过程是以图 12 的流程图中的优选方式实现的话，则经过修改的树可能包含少于 29000 条指令，比数字此结构可检索的单词对象的数目要少。此结果值得注意并且很新颖，因为对于一个键击每个指令的反应仅仅是通过后接单个符号只改变对象列表 430 中的一个对象。这是折叠树和重复使用公共指令序列作为通用对象构造规则的检索软件产生的结果。

在图 9 中示出的节点是实例。节点 560 具有两个有效键，在其有效键字段 562 “010100000”中以“1”表示。在优选实施例中“1”的位置表示第二个和第四个键是有效路径并且具有通到与其相关联

的子节点的指令包和指针 566 和 568。包 566 包含 3 个指令“(1, 1, 0)”,“(1, 2, 0)”和“(2, 2, 2)”,前后的指针“P”链接 560 到子节点。如子例程 600 已经处理过通往子节点 560 的键击列表,则子例程 620 被调用来处理“2”键(该键在本实施例中是 ABC 键),之后发生下面的过程。指令 561 将把键 ABC 的第一逻辑符号(“a”)后接于索引 1 中的老对象以便建立索引 1 中的新对象。561 的第三字段,“0”,是 STOP-FLAG 557 的伪值,表示这不是当前包的最后指令,所以对下一个指令 563 进行解释。指令 563 将把键 ABC 的第一逻辑符号(“a”)后接于索引 2 中的老对象以便建立索引 2 中的新对象。新对象的索引将是 2,因为新对象的索引是隐含地以指令本身的顺序建立的,比如第二指令永远建立第二对象。563 的第三字段,“0”,是 STOP-FLAG 557 的伪值,所以对下一个指令 567 进行解释。指令 567 将把键 ABC 的第二逻辑符号(“c”)后接于索引 1 中的老对象以便建立索引 3 中的新对象。567 的第三字段,“1”,是 STOP-FLAG 557 的真值,表示这是当前包的最后指令,所以子例程 620 的执行将从步骤 638 转到步骤 644。

可以将两个或更多的包含不同指令包 408 的节点组合成为单个节点,该节点可用于与多个节点独立时一样的的目的,这就意味着词汇树 110 中的某些节点在新的意义上是冗余的。对于本发明单词“冗余”用于两个节点的含义是通过示于图 14-17 中的优选实施例中的软件过程的操作可以将一个节点去掉。

比如,比较图 9 中的节点 560 和 574。键 2 的指令包 566 和 571 完全相同,但节点 560 的键 4 上的指令 570 与节点 574 的键 4 上的指令 572 矛盾,两者不能互相代替,两个节点也不能组合成为一个来完成两者的工作。下面比较节点 560 和 576。与每个节点上的键 2 相关联的指令包 566 和 577 完全相同。指令包 569 和 578 的 STOP-FLAG 字段 557 的设置不同,但该差异不会使它们发生矛盾。图 14 中的子例程 620 的对象检索过程的根本结果是通过给定键在一个节点执行一组指令生成的新对象列表。附加的对象可后接到对象列表的尾

部而不会对正确处理该节点的任何子节点造成损害。这样，在 578 之后执行附加的指令不会在节点 576 的子节点的处理中造成错误。此过程的本质只有在执行错误的指令或执行的指令过少才会遭到破坏。与此类似，在节点 576 的键 9 上存在有效键不会与在节点 560 中缺少键 9 矛盾。因此，节点 560 和 576 是冗余的，并且可以合并成为一个新节点 582，该节点可达到两者的净效应，并且其作用与两者的子节点的父节点相同。可以理解，在确定冗余度上指针也有作用。在一个优选实施例中，在树中的与不继续形成更长单词的词干的单词相关联的键击序列的最后键击中在有效键包 408 中的指针具有特殊值“NULL(零)”，这表示再没有子节点了。此类节点称为“端节点”。对于具有对两个节点是共通的有效键的子节点上的两个节点，相应的子节点必定是冗余的，因为其父节点是冗余的，并且对从子节点下行的节点也同样，一直到到达端节点或对所比较的节点共通的有效键序列上不再有后代为止。

图 14 至 17 示出为用于压缩和折叠压缩类似于图 10 示出的树 110 的词汇模块树的软件过程的优选实施例的流程图。图 14 为用于建立压缩词汇模块的软件过程的优选实施例的流程图。在步骤 652 中，扫描词典以认定在如图 6 中针对法文词汇模块所示的物理键上出现的符号以外的任何必需的附加多义性符号。在步骤 654-656 中，对各符号在其相应的各键上依照其在输入的词典中的使用频率的降序赋予其逻辑索引，与图 7 示例中的相同。如给定具有频率的对象词典，本行专业人士很清楚步骤 658 如何建立 110 形式的词汇树。在步骤 660 中，认定冗余节点并将其合并以减少数据的重复并从而将与单个对象相关联的孤立的指令序列转变为用于检索多个对象的通用规则。此过程示于图 15。步骤 662 从端节点中认定所有保持为 NULL(零)的指针并将其改变为指向具有最大父节点数的节点，从而增加模块中的规则数。可以理解，也可以使用其他规则来赋予子节点 NULL 指针，并且此种规则可根据正在处理的键击的相关因子在对象检索时动态地应用。在步骤 664 中，计算剩余的各独特指令 558

及指针 404a, 以使其可以编码为独特的位图而将较短的位图赋予较高频率的指令和地址以节省空间。在本发明的一个优选实施例中使用 Huffman 编码来将最小长度的位图赋予指令和地址。可以理解, Huffman 编码技术是业内公知的, 所以毋庸赘言。另外, 是多个父节点的子节点的节点可以以特殊的顺序存储以便可快速检索和减少其定址所需的位(比特)数。最后, 将如上示出的数据存储于步骤 666 中的文件内。

在步骤 658 中建立树的同时选择用来描述待存储的对象的指令 558 中间, 可以理解, 当对象是单词或单词词干时, 其字符序列包含适宜用来增加树 110 中的节点的冗余的附加数据。比如, 英文中并非所有的字母对同样常见, 比如“s”通常与“t”配对。字母对, 或称双字母组, 统计学可用于从前一个字母预见对象中的最可能的下一个字母。采用这种预见时, 在逻辑符号索引表 550 中的多义性符号的逻辑顺序可动态地改变以进一步优化第一位置的使用。这种预见可扩展到三个字母, 三字母组, 并且广而言之可扩展到  $n$  字母组。

图 15 为折叠词汇模块的树 110 的软件过程的优选实施例的流程图。从图 14 的流程图的步骤 660 进入步骤 670。步骤 672 从图 10 的树 110 根节点 111 下的第一个节点开始启动过程。步骤 674 调用示于图 16 中优选实施例的流程图的子例程 690 来定位与当前节点冗余度最大的节点, 如该节点存在。如目标节点找到, 则判定步骤 676 将过程指向步骤 678, 在该步骤中合并冗余节点, 取消树中的重复数据, 将多个独立的指令归类为共享序列, 成为可用于将键击序列与对象关联通用规则。如判定步骤 676 失败, 则判定步骤 680 检查过程是否完结。如尚有更多的节点要处理, 流程就转到步骤 682 去认定另一节点。

图 16 为用于在树 110 中找寻与给定节点相比较具有最大冗余度的节点的软件过程的优选实施例的流程图。从图 15 的流程图的步骤 674 进入步骤 690。步骤 692 启动一个位置标志符 MAX-SAVING 用于

测得的冗余度。步骤 694 从图 10 的树 110 根节点 111 开始启动过程。步骤 696 调用示于图 17 中优选实施例的流程图 710 来计算给定节点与当前节点的冗余度。判定步骤 698 检查冗余度是否大于报告的 MAX-SAVING。如是，步骤 700 记录临时发现的节点的身份 BEST-NODE 为相对给定节点冗余度最大的节点并且报告的冗余度定为 MAX-SAVING。判定步骤 702 检查是否所有的节点都已经评估完毕。如否，则流程继续转到步骤 704，在该步骤中从当前节点转向下一个节点。从步骤 704 流程返回到步骤 696。如在判定步骤 702 中检查结果是最最后一个节点已经评估，则步骤 706 将节点的身份作为冗余度最大者，如有，则返回给给定节点转到图 15 的子例程 670 的步骤 674。

图 17 为用于计算两个特定节点之间的数字冗余度的软件过程的优选实施例的流程图。从图 16 的流程图 696 进入步骤 710。步骤 712 初始化重复指令的计算。步骤 714 将 KEY-INDEX 初始化为 1。步骤 716 从指定作为参数的两个节点 400 之中的第一个将与键 KEY-INDEX 相关联的指令包 406 读出到子例程并将其置于临时列表 LIST-A 中。如键 KEY-INDEX 不是有效键，不读指令。步骤 718 从指定作为参数的两个节点 400 之中的第二个将与键 KEY-INDEX 相关联的指令包 406 读出到子例程并将其置于临时列表 LIST-B 中。如键 KEY-INDEX 不是有效键，不读指令。判定步骤 720 判断 LIST-A 和 LIST-B 是否有的为空。如否，步骤 722 从 LIST-A 和 LIST-B 各取出一条指令并将各自的剩余指令数减 1。判定步骤 724 检查在其 LOGICAL-SYMBOL-INDEX 和 OBJECT-LIST-INDEX 字段中指令是否相同。如否，在步骤 726 中向子例程 690 的步骤 696 返回一个失败代码表示无冗余。如判定步骤 724 的判断为“是”，则步骤 728 将计数 SAVED-INSTRUCTIONS 增加 1。控制又转到步骤 720。如步骤 720 的检查是“真”，则控制转到判定步骤 730，该步骤检查是否两个节点的所有可能的键都已经比较过。如否，步骤 732 使 KEY-INDEX 加 1，并且控制转到步骤 716。如步骤 730 的判断是正，则控制转到步骤 734 重置 KEY-INDEX 为 1。判定步骤 738 检查是否两个指针有的为空

(NULL)，端节点或任何无效键的指针为空。如都不空，控制转到步骤 740，该步骤利用子例程 710 分别检查由两个不空指针指向的子节点是否为冗余。步骤 740 的结果在判定步骤 742 中检查。如两个子节点不是冗余的，在步骤 744 中返回一个失败代码。否则认为两个子节点为具有某一数字分数的冗余节点，该数字由步骤 746 累积。判定步骤 748 检查是否与最后键(在优选实施例中的键 9)相关联的指针已经检查过。如否，步骤 752 将 KEY-INDEX 增加 1 并将控制转到步骤 736。如在判定步骤 748 中的检查判断所有的指针都已经检验过，则在子例程进入步骤 710 时将原来认定的两个节点的累积冗余度值返回。

可以理解，冗余度的数值计算可加权以便考虑附加的因素，如在每个节点的分支数及作为子节点指向节点的父节点数。可以理解，如两个节点因为与某一个键相关联的指令的排序而不是冗余的，则可以将与输入词典中的低频单词相关联的指令重新排序而不会影响与较高频率对象相关联的指令的优先级，从而可增加树的冗余度。

### III. 代表性系统的操作

图 18A - 图 18B 示出在代表性应用简化键盘多义性消除系统中的便携式计算机 52 的显示器 53。在打开便携式计算机的电源之后，文本区 66 和选择列表区 70 是空的。在图 18A 中，用户打入短语“*This is a test*”。词汇模块认定最后 4 个键击 TUV, DEF, PQRS, TUV 的最可能的解释是单词“*test*”并且将此解释临时放在插入点 900，而且还将此单词放在选择列表中作为默认解释 901。选择列表中的此第一条目周围有一个实线框(表示其为隐含选择对象)，并且也临时放在文本区的插入点 900 处，周围也带有实线框。词汇模块也认定此单词的解释“*vest*”及词干解释“*vert*”，并将这些解释放在选择列表中。图 5A 和图 5B 中示出的标点键处理软件将文本对象“*test*”存储于 *curPunctWorld* 缓冲区中。

在图 18B 中，用户随后按下“-”标点键 63。在数据库中并没有对

象与此后面带有标点键 63 的 4 个键 TUV, DEF, PQRS, TUV 的键击序列相匹配。这样, 出现在选择列表中的只有几个文本对象是由标点键处理软件添加的, 包括 curPunctWorld 缓冲区中的“test”后接句点 911, 短划 913 和撇号 914。结果形成的默认选择列表对象是“test.” 911, 该对象也临时放在文本区的插入点 910 处。

在图 18C 中, 用户其后按下空格键 922。这生成一个显式空格字符, 它结束当前的键击序列并使系统接受临时放置的文本“test.”。也输出一个后面的空格, 之后系统光标 920 出现。键击缓冲区现在是空的, 并且当前选择列表 921 也是空的。这显示了标点键 63 的一种默认行动-当在激活空格键(或选择键)之前激活时, 如在包含标点字符的数据库中没有相匹配的对象, 则原来的默认文本对象带一个句点输出。

在为英文(标志 Apostrophe\_S 设定为“真”)设计的系统场合下, 图 18D 示出在图 18B 的情况下随后用户按下 PQRS 键 932 的结果。这将引起图 5B 中的步骤 322 至 342 的处理过程, 并导致标点键处理软件按以下顺序在选择列表中添加“test’ s”, “test-s”, “test-r”, “test-p”和“test-q”。这样, 得到的默认选择列表对象是“test’ s” 931, 该对象也临时放在文本区的插入点 930 处。

图 18E 示出在图 18D 的情况下随后用户按下 TUV 键 932(意图是打入带短划的单词“test-run”)的结果。这将引起图 5B 中的步骤 332 至 340 的处理过程, 并导致标点键处理软件“test-”输出作为所接受的文本, 并且键击缓冲区修改为只包含最后两个键击 PQRS 及 TUV。根据这两个键击对数据库进行查询(在图 5B 的步骤 360 中)得到如图 18E 所示的结果。文本对象“st”是频率最高的, 所以也就作为默认选择文本对象“st” 941 出现, 并且也临时放置文本区的插入点 940 处。

图 18F 示出在图 18E 的情况下随后用户按下 MNO 键 952 以结束打入带短划的单词“test-run”)的结果。不会发生特别的标点键处理, 因为在图 5B 的步骤 362 中在处理前面的键击时所有的标志都已经清

除。于是系统就单纯地处理当前键序列 PQRS, TUV, MNO。根据这些键击对数据库进行查询得到如图 18F 所示的结果。文本对象“run”是频率最高的, 所以也就作为默认选择文本对象“run” 951 出现, 并且也临时放置文本区的插入点 950 处。

图 18G - 图 18K 示出在为法文(标志 Apostrophe\_Term 设定为“真”)设计的系统场合下的示例。在词示例中, 用户打入文本对象“c' est”该对象实际上是“c”与单词“est”的连接。图 18G 示出用户按下 ABC 键 962 的结果。选择列表表示出与法文 ABC 键相关联的单个字符对象, 包含带重音符的 à, â 和 ç, 不会发生特别的标点键处理, 因为所有相关的标志在前面都已经清除。于是系统就单纯地处理当前键序列 ABC。根据这些键击对数据库进行查询得到如图 18G 所示的结果。文本对象“à”是频率最高的, 所以也就作为默认选择文本对象“à” 961 出现, 并且也临时放置文本区的插入点 960 处。

在图 18H 中, 用户随后按下.-' 标点键 63。在数据库中存储有对象“c' ”和“a-”, 并在图 3 的步骤 156 至 160 中在查询数据库之后添加到选择列表中, 结果得到的默认选择文本对象是“c' ” 971。结果, 进行图 5B 中的检查 300 至 306, 并且在步骤 308 将 pState 标志设定为 PUNCT\_APOS。在选择列表中的“c' ” 971 和“a-” 973 之后, 标点键处理软件(在图 5B 中的步骤 312, 314 和 316 中)添加由 curPunctWorld 的“à”与后接句点 974, 短划 975 和撇号 976 构成的对象“à.”, “à-”和“à' ”。得到的为默认选择文本对象是“c' ” 971, 并且也临时放置文本区的插入点 970 处。

在图 18I 中, 随后用户按下 DEF 键 982。因为由于处理前面图 18H 的键击的结果使 pState 标志设定为 PUNCT\_APOS, 所以标点键处理软件执行图 5A 中的步骤 274, 276 和 262。输出当前默认选择列表对象“c' ”作为被接受的文本 983, 并且系统简单地处理得到的单键序列 DEF 作为添加当前键到清除的键击缓冲区的结果。根据这些查询数据库, 此键击得到的结果是图 18I 中示出的选择列表。文本对象



“d”是频率最高的，所以也就作为默认选择文本对象“d”981出现，并且也临时放置文本区的插入点980处。

图18J至图18K示出随后用户按下PQRS键992(在18J中)和TUV键997(在18K中)以结束打入单词“est”的结果。不会发生特别的标点键处理，因为在图5A的步骤262中在处理前面的键击时ClearPunctMode标志已经设定，所有标点键的缓冲区和标志都已经在图5A的步骤202中清除。于是系统就单纯地处理当前键序列DEF, PQRS, TUV。根据这些键击对数据库进行查询得到如图18K所示的结果。文本对象“est”是频率最高的，所以也就作为默认选择文本对象“est”996出现，并且也临时放置文本区的插入点995处。

法文包含很多很常用的冠词，前置词和代词，如le, ne, ce, se, je, que, tu和te，这些词在置于以元音开始的单词的前面时，将其元音改变为撇号并且打字时与后面的单词之间不留空格。如果没有在上边的示例中示出的标点键63支持的动作，这种经常发生的单词对象的结合就会要求用户另外激活选择键以便在打入第二对象(est)之前结束第一个单词对象(在示例中的c')。这样，标点键63支持结尾为撇号的单词自动连接的功能可显著提高像法文这样的语言的系统效率及自然性。

虽然本发明的优选实施例已经示出和描述，但应该理解，在不脱离本发明的精神和范围的条件下可进行各种改变。比如，本行的专业人士将会理解简化键盘多义性消除系统的键盘54的数据键数目可少到3个多到20个。此处公开的多义性消除技术同样可应用于不同尺寸的键盘。与标点键63相关联的特殊字符可改变。比如，本行的专业人士将会理解在某些应用中句点可代之以逗点。另外，在不同的上下文中生成什么标点字符的选择也是可以改变的。比如，连续两次激活标点键63后再按下a可生成长破折号“—”，而连续3次激活标点键可生成“...”。

本行的专业人士也将会认识到在计算机中可常用附加的词汇模块，比如包含法律名词，医学名词，以及外语名词的词汇模块。通

过系统菜单，用户可配置系统使附加的词汇单词在可能的单词列表中首先出现或最后出现，采用特殊的彩色或加亮。结果，在后附的权利要求书中，可以理解本发明可以通过与此处所专门描述的方式不同的方式实现。

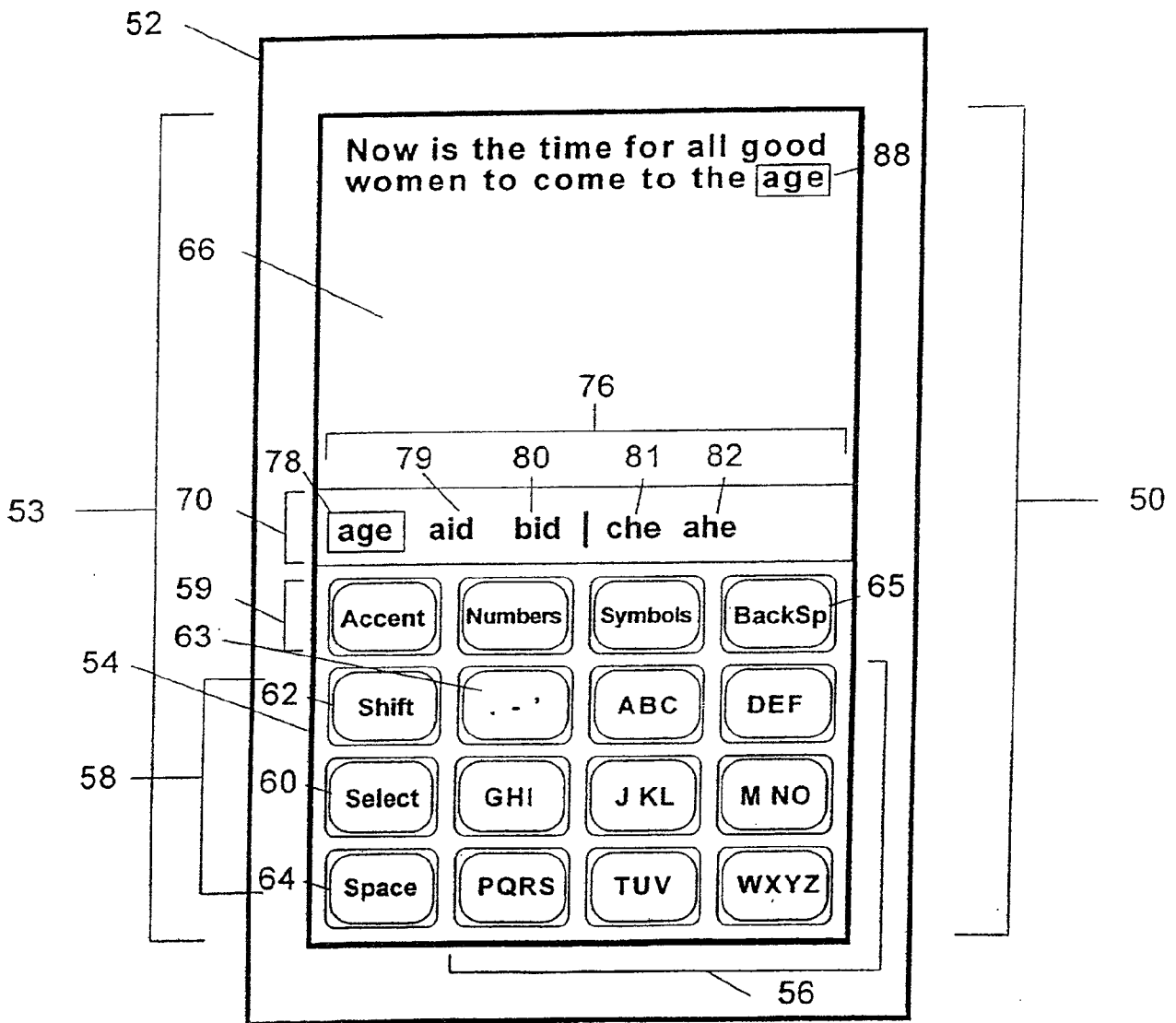


图 1

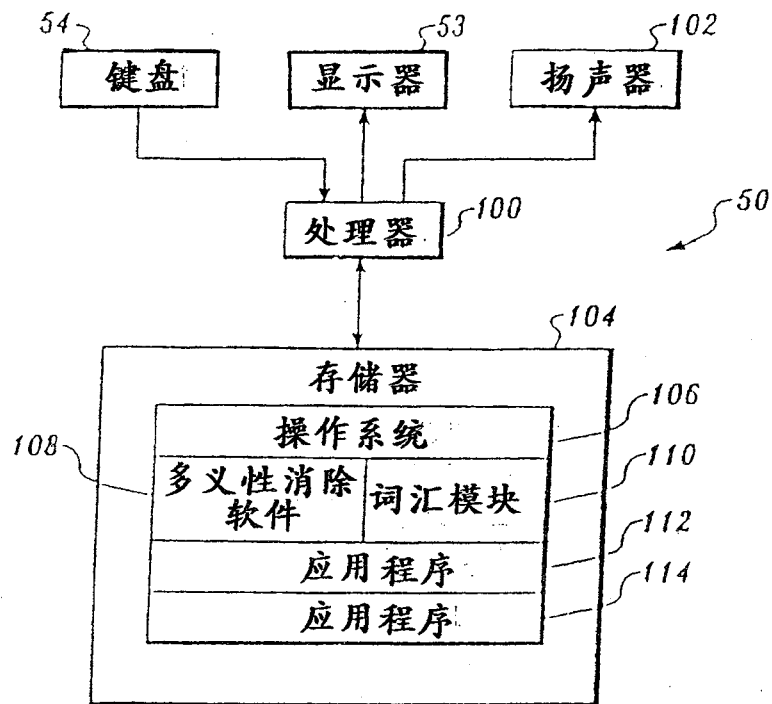


图 2

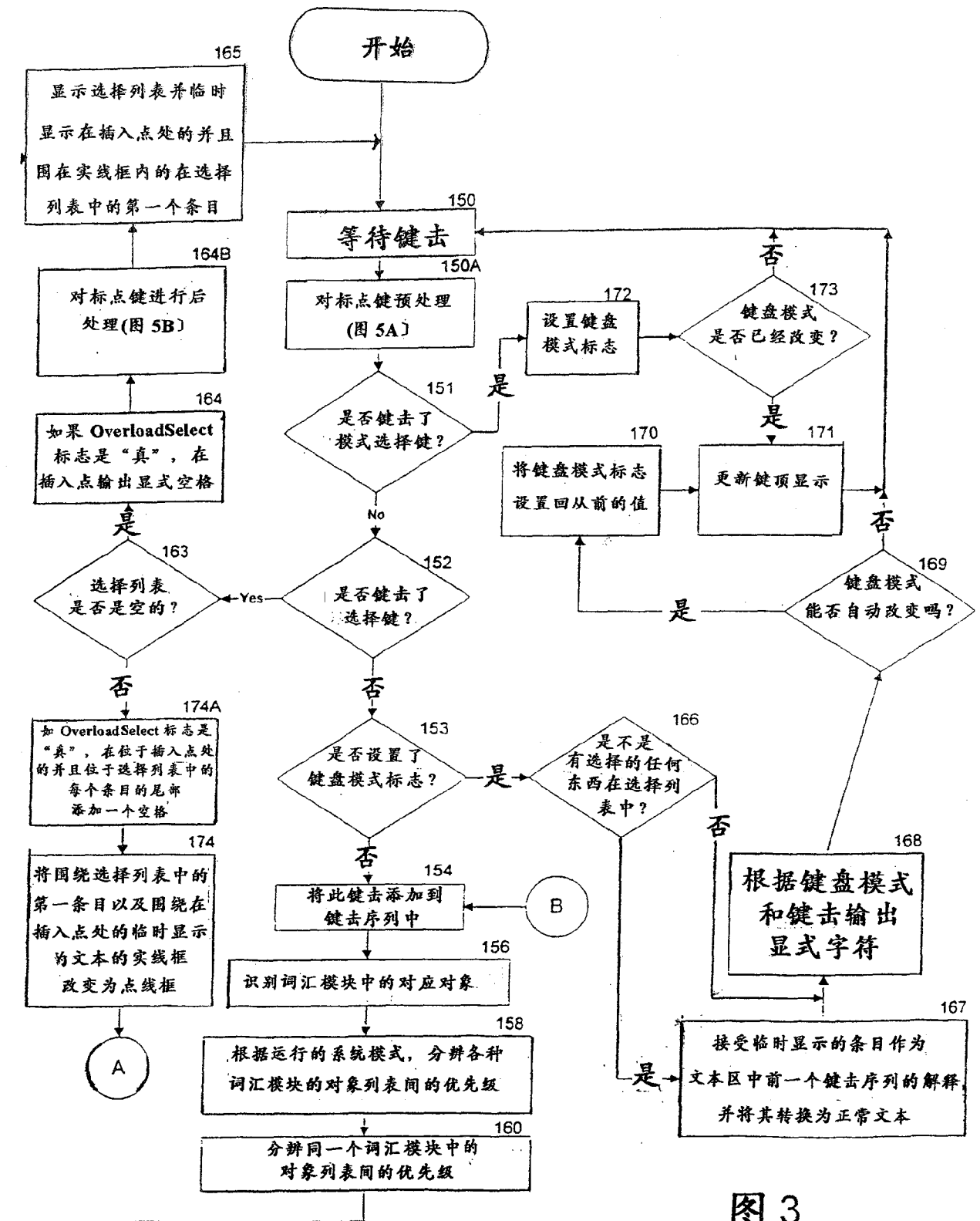
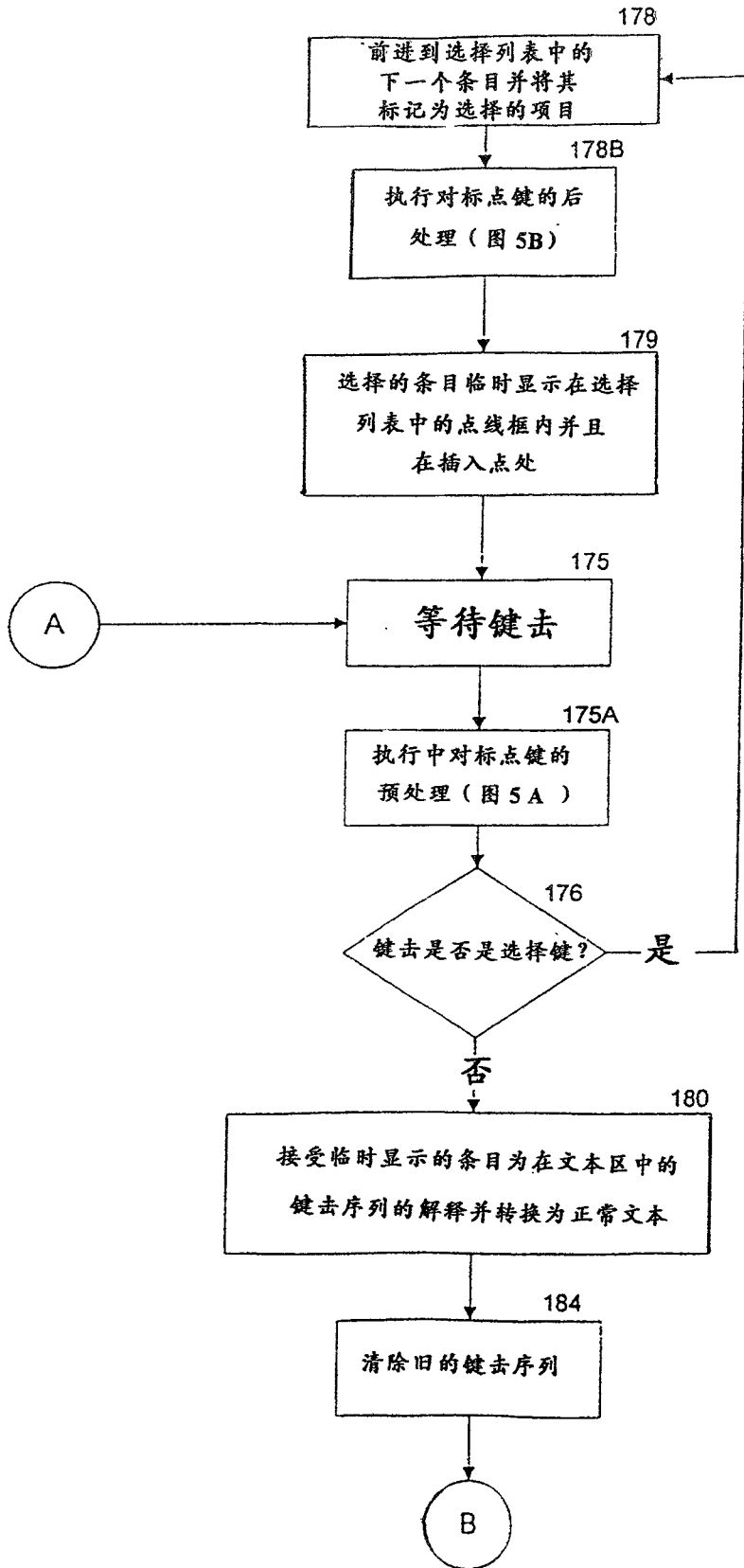


图 3

图3 (续)



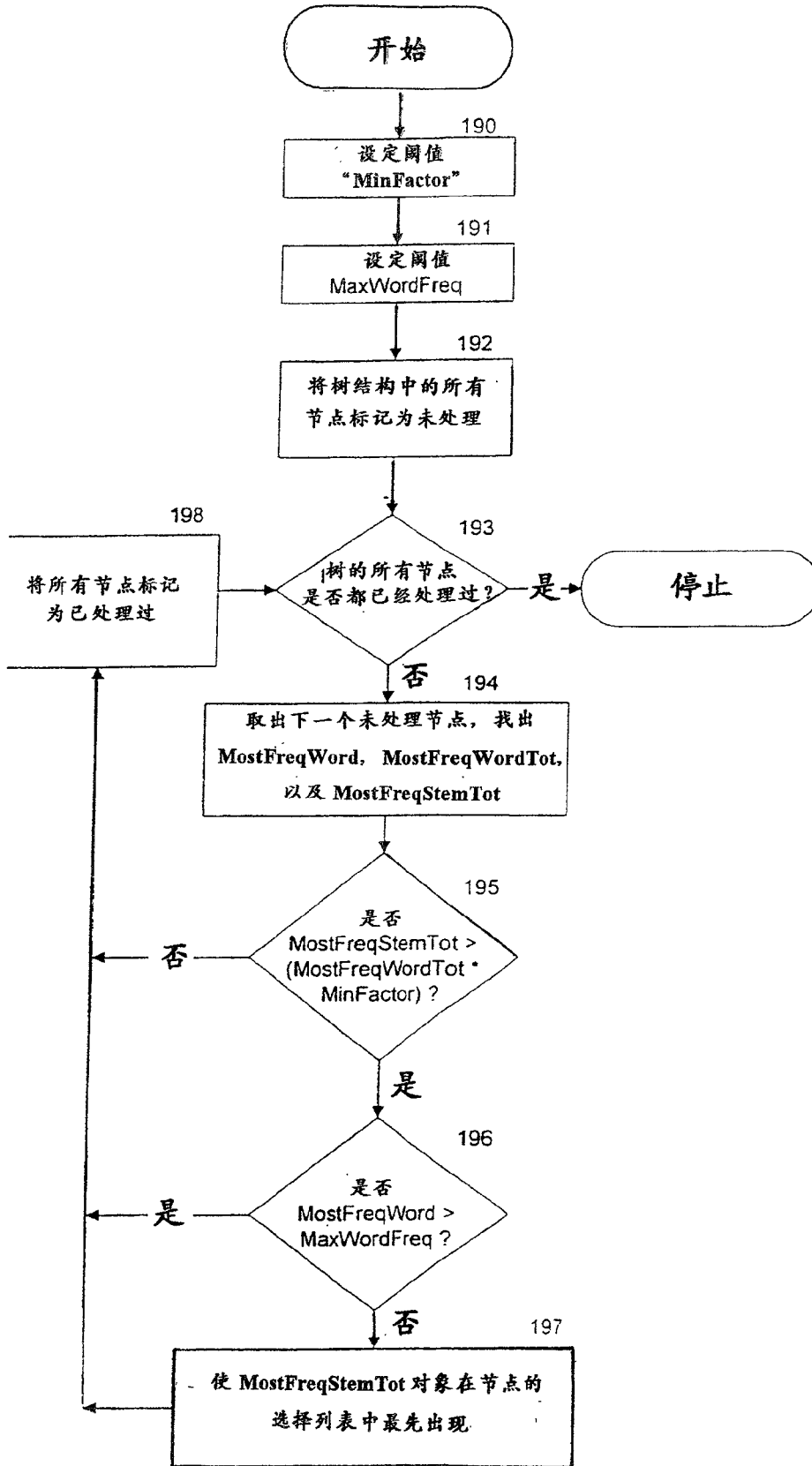


图 4

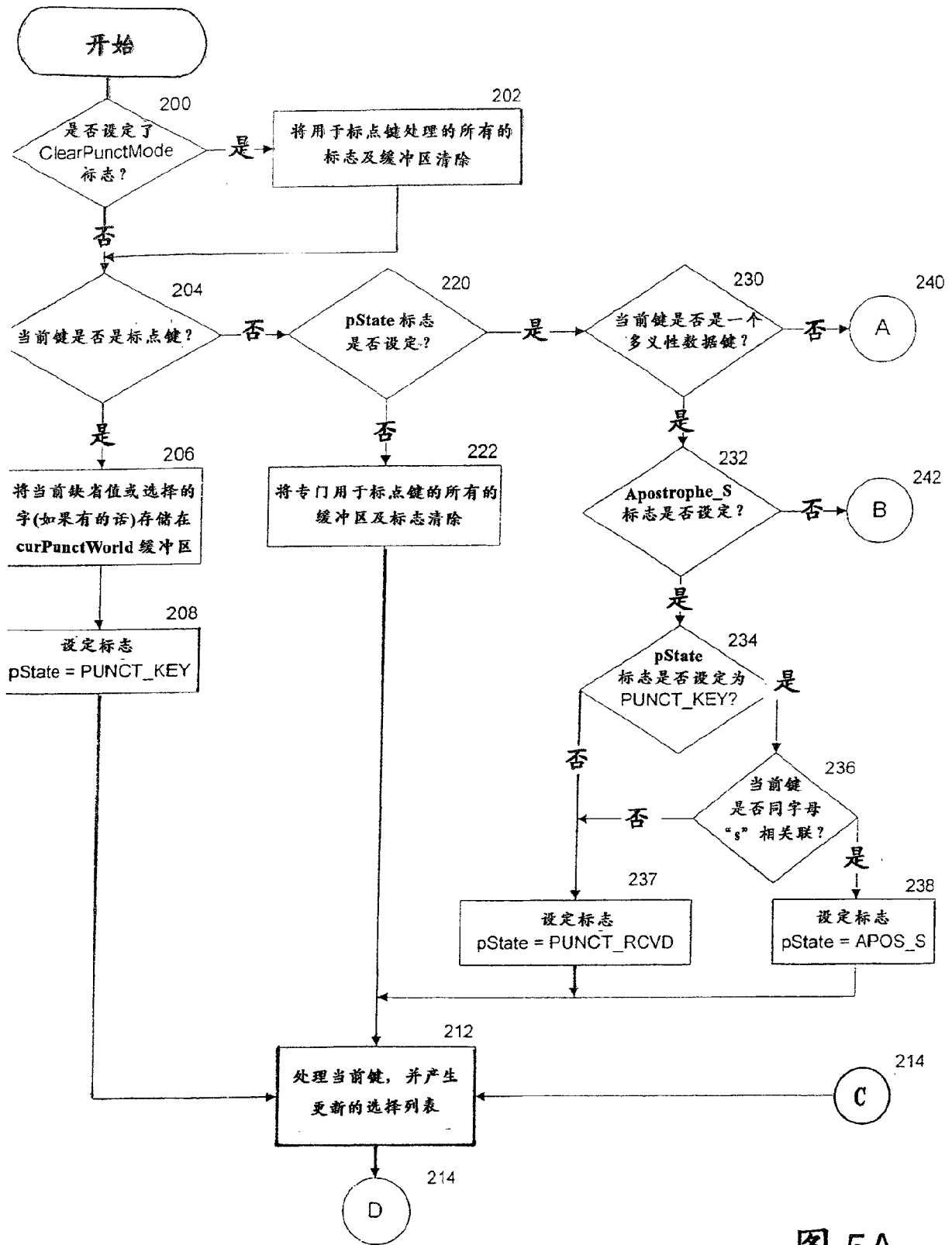


图 5A



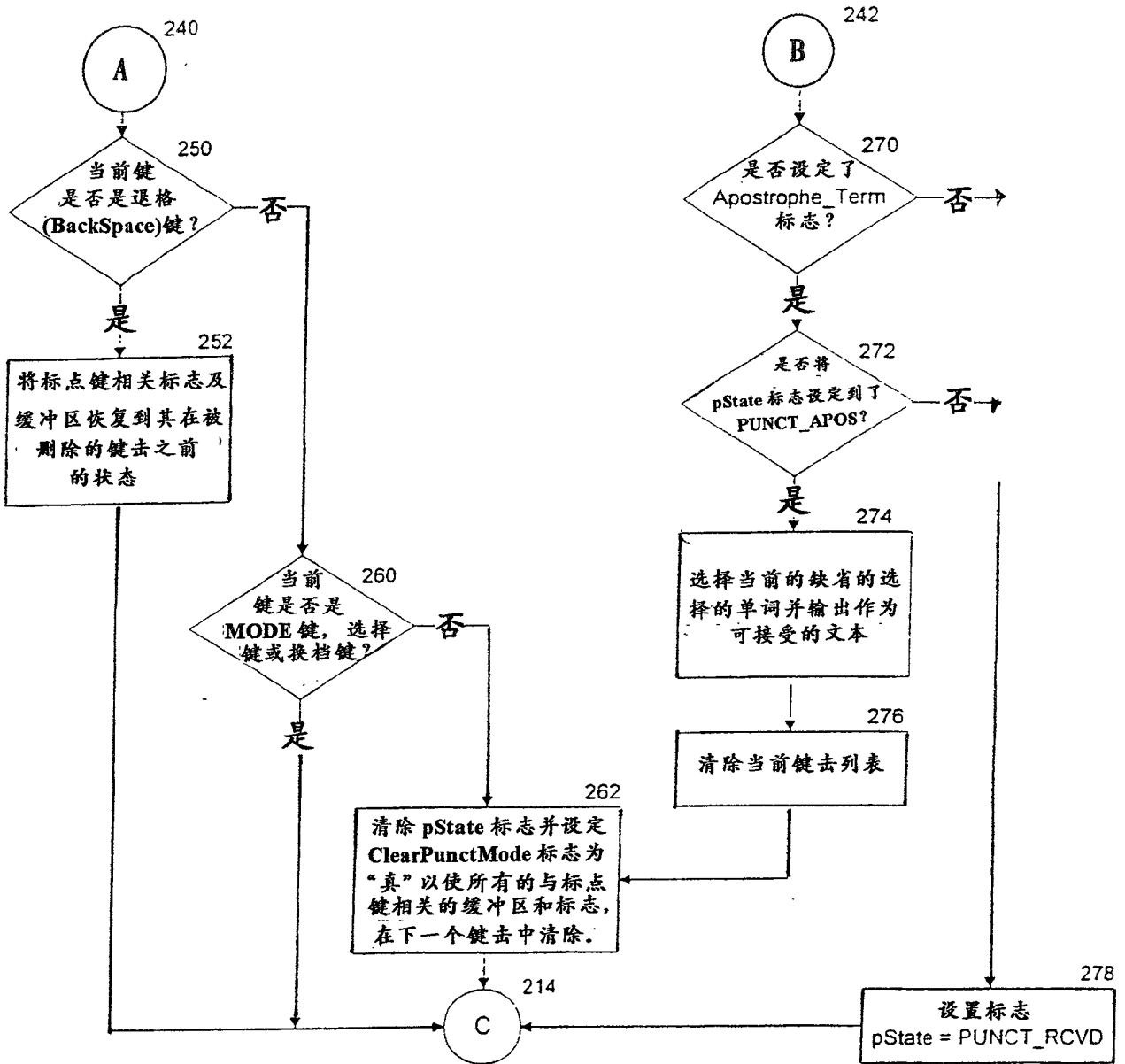
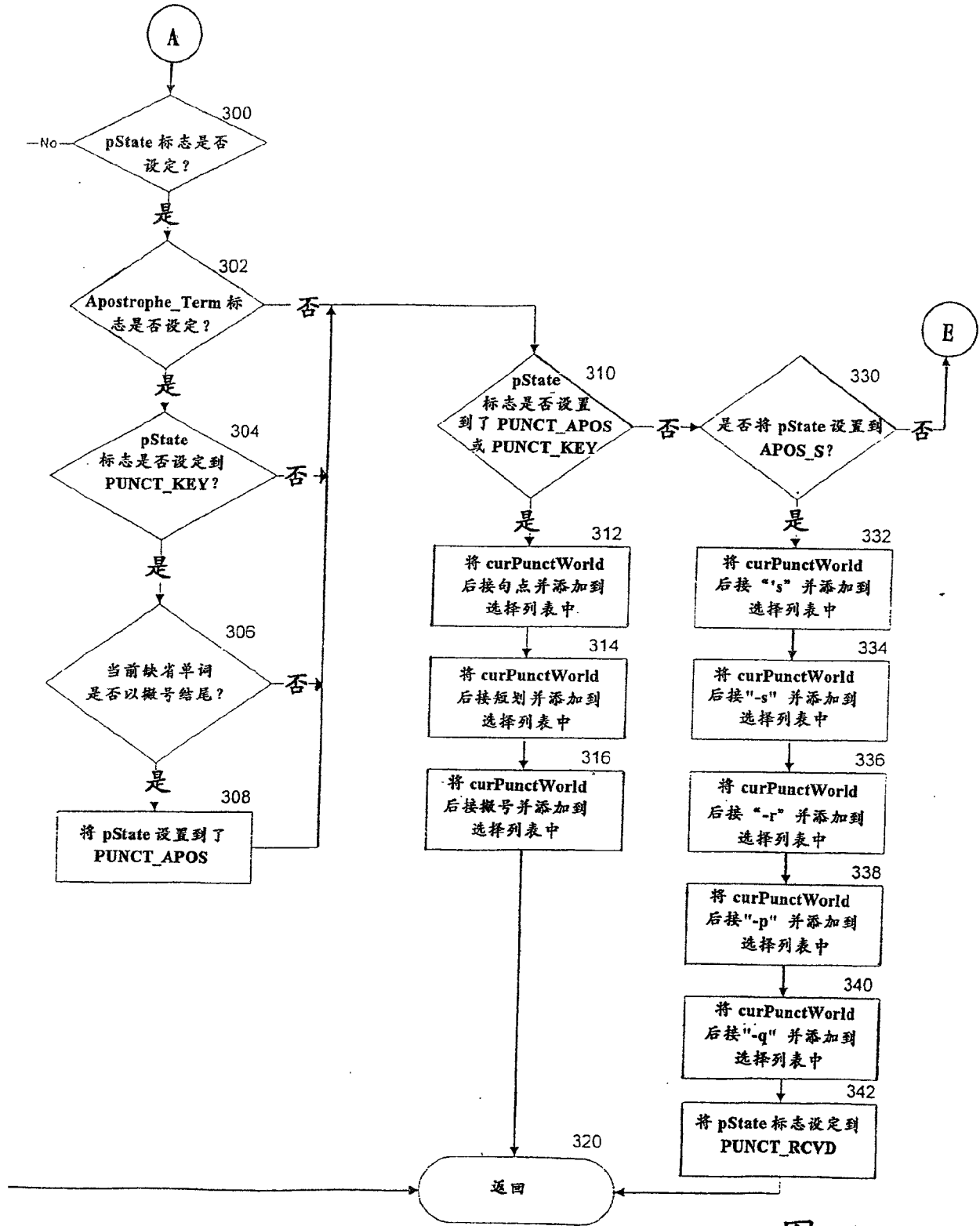


图5A(续)



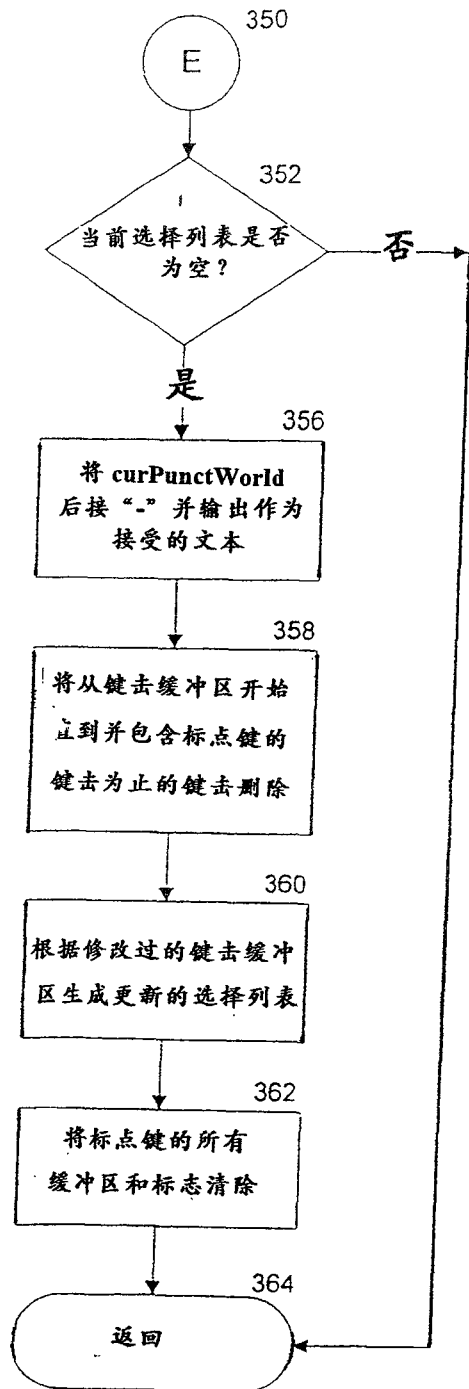
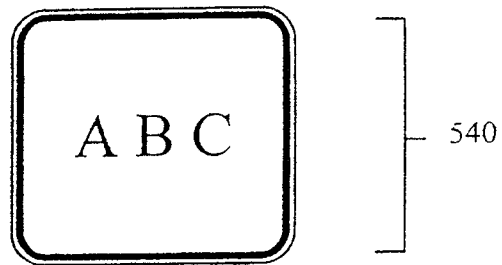


图 5B(续)

图 6

物理键



逻辑键(法文)

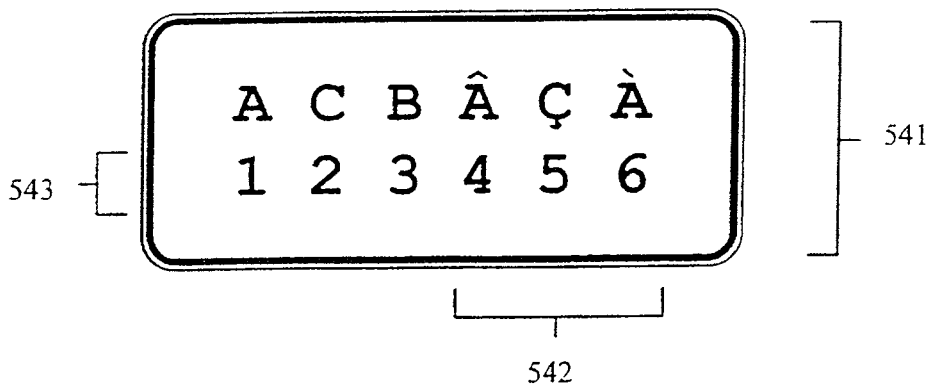


图 7

## 法文的逻辑符号索引表

	1	2	3	4	5	6	7
1	.	-	'				
2	a	c	b	â	ç	à	
3	e	é	d	f	è	ê	ë
4	i	g	h	î	ï		
5	l	j	k				
6	n	o	m	ô	œ		
7	r	s	p	q			
8	t	u	v	û	ú	ù	ü
9	x	z	y	w			

550

551

552

图 8A

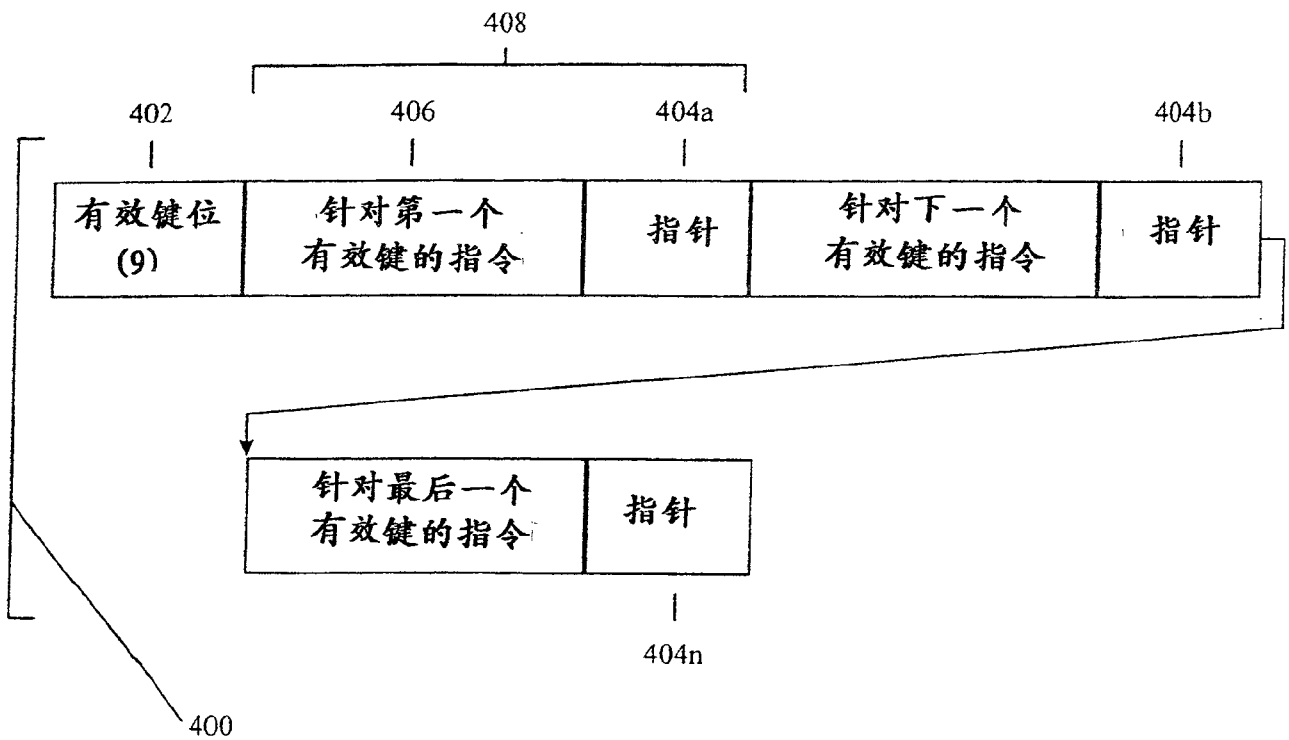


图 8B

指令语义

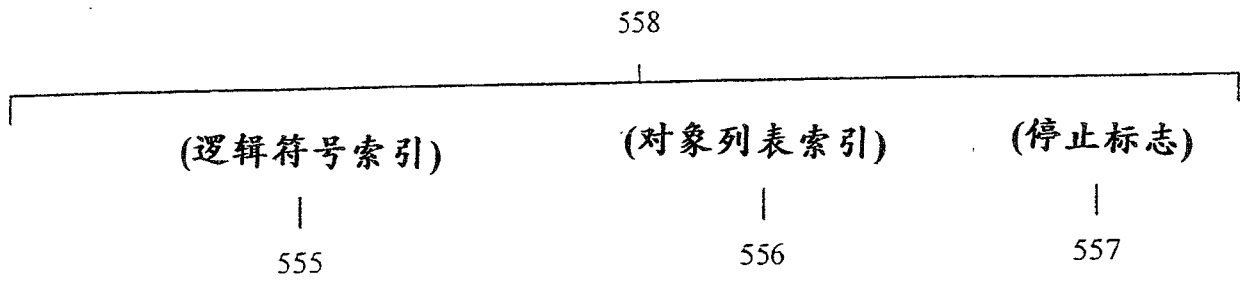


图 9

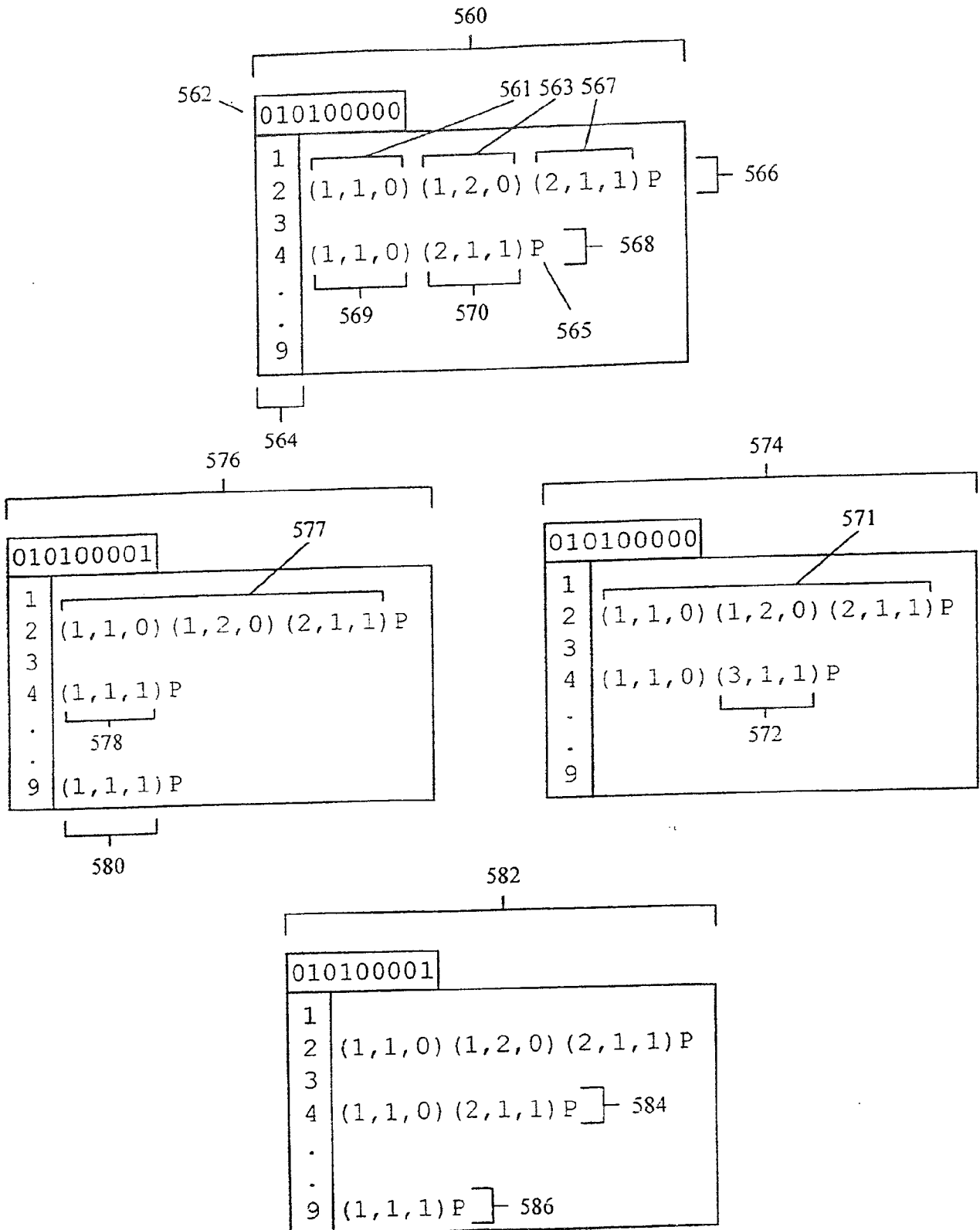




图 10

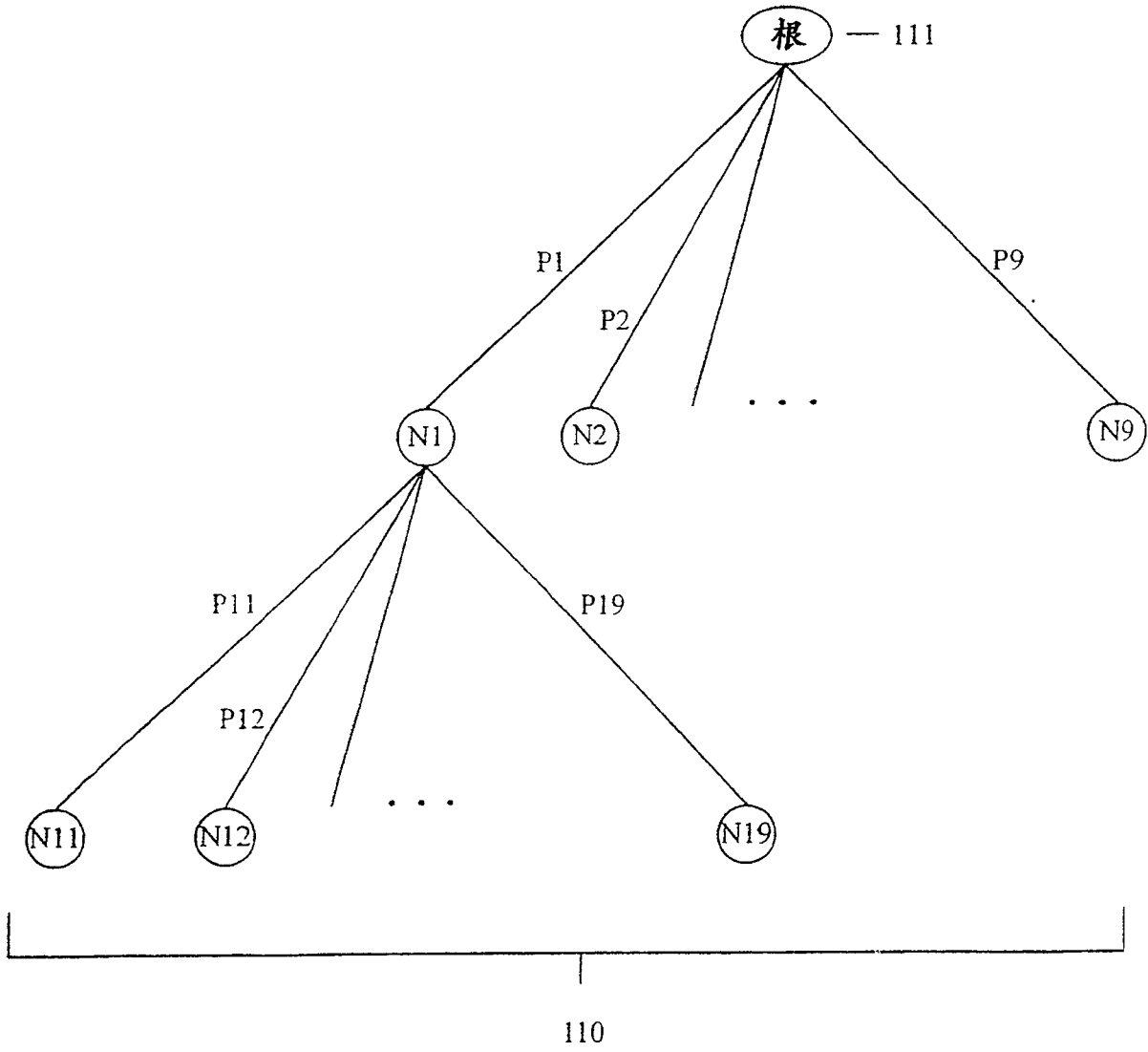
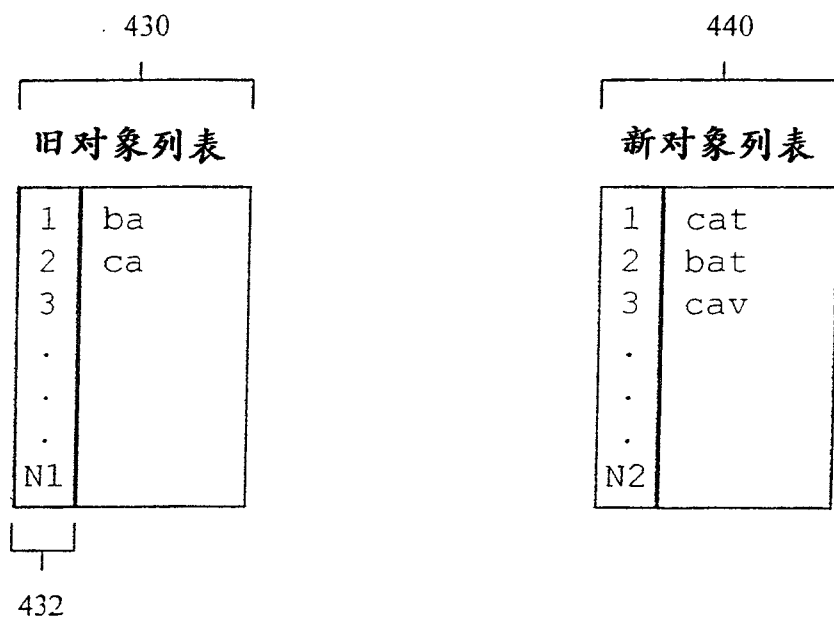


图 11



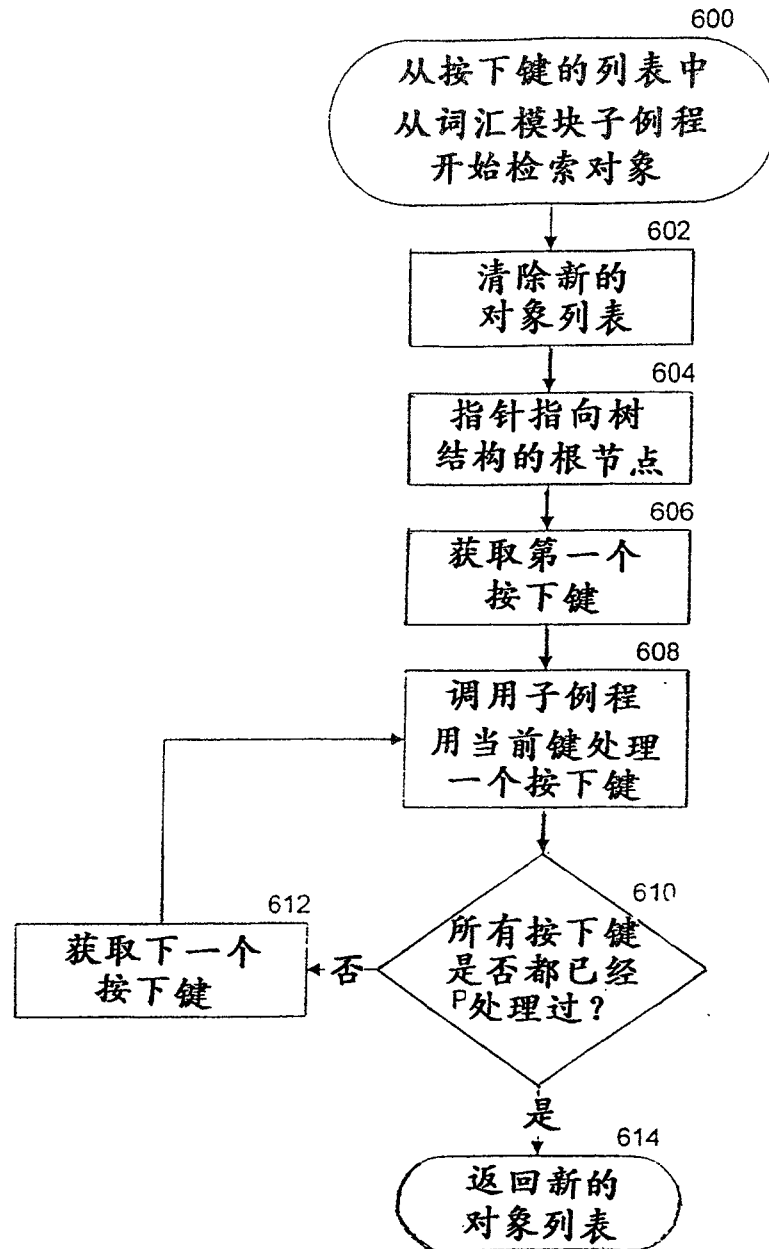
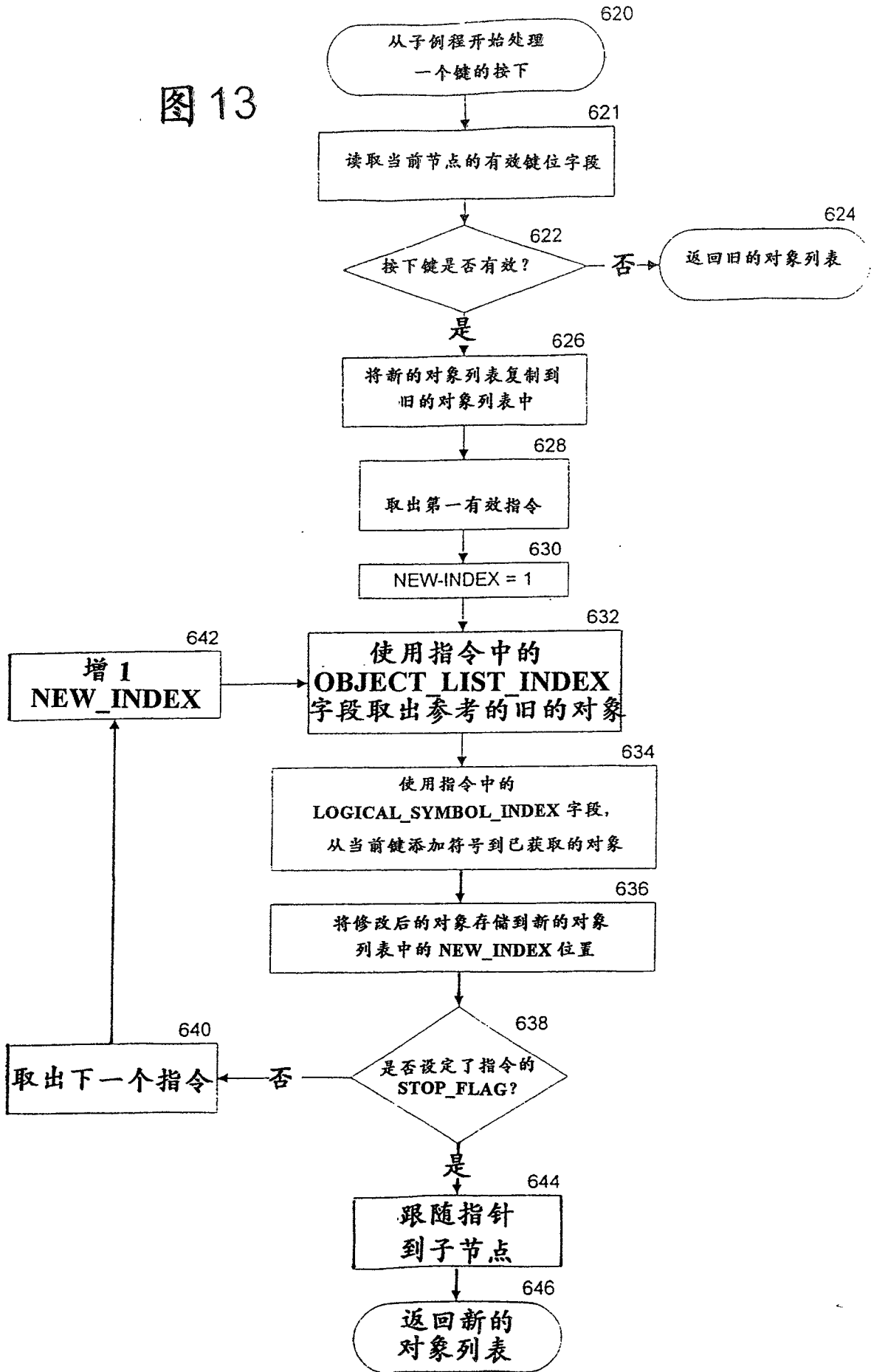


图 12

图 13



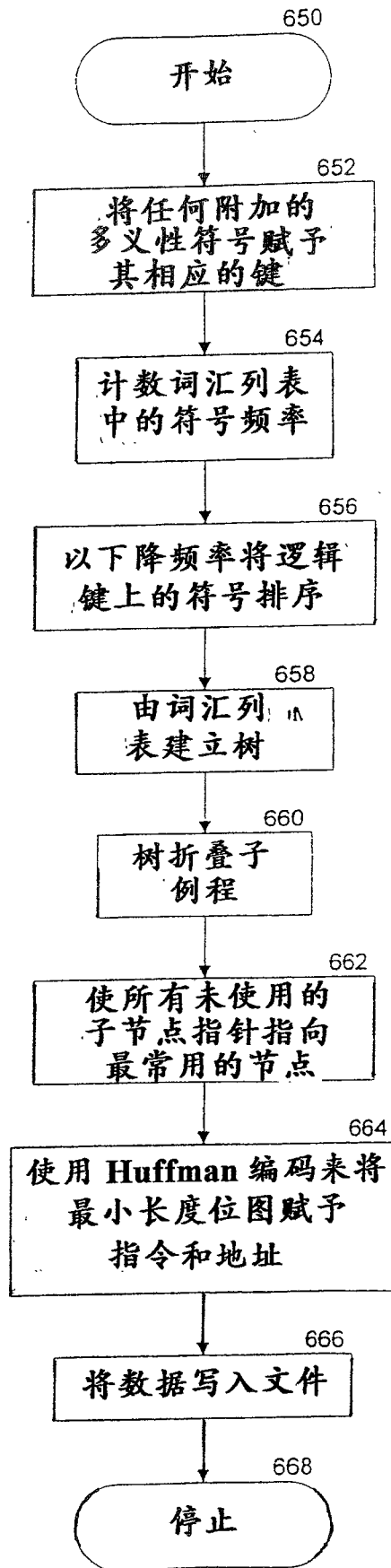


图 14

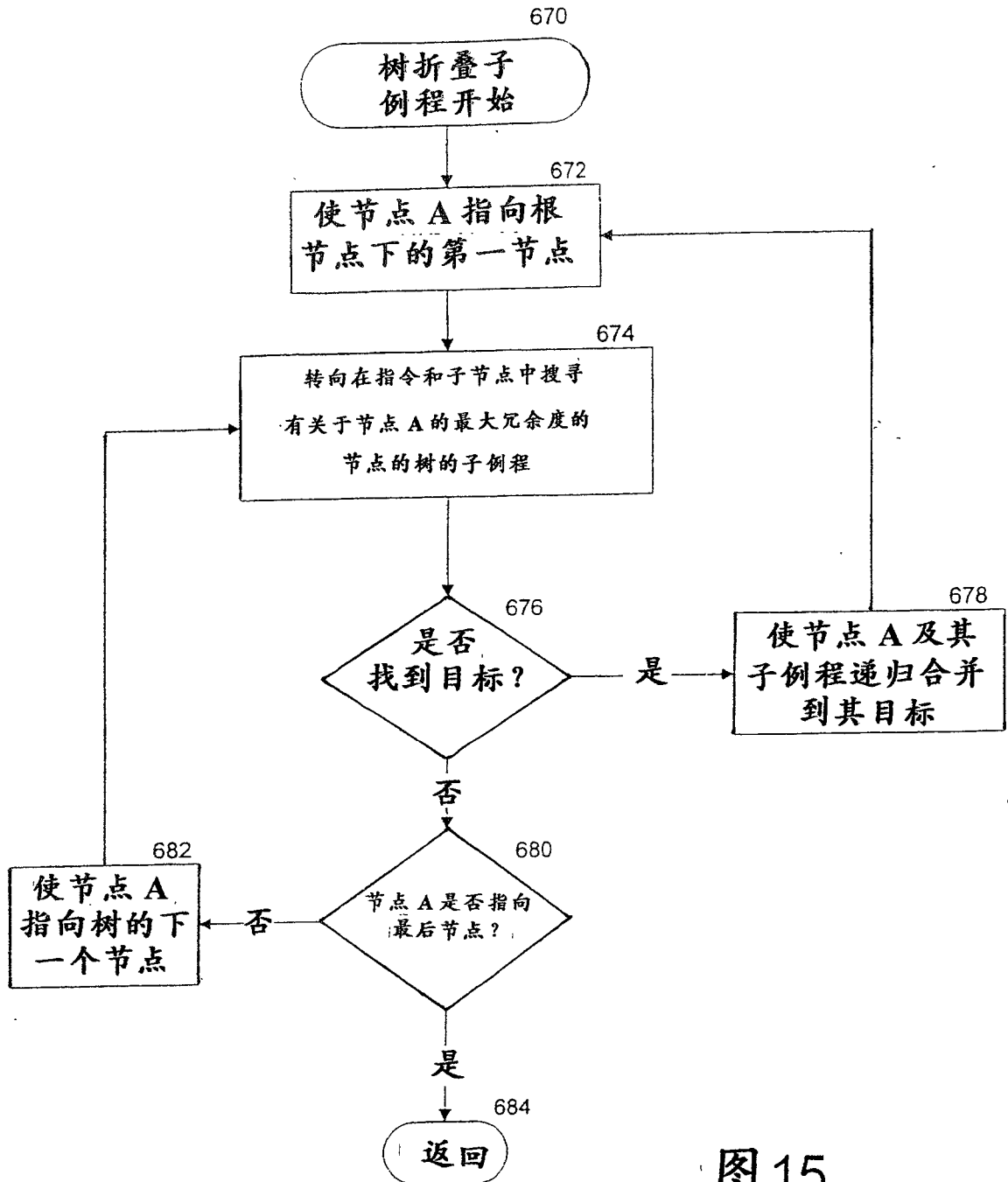


图 15

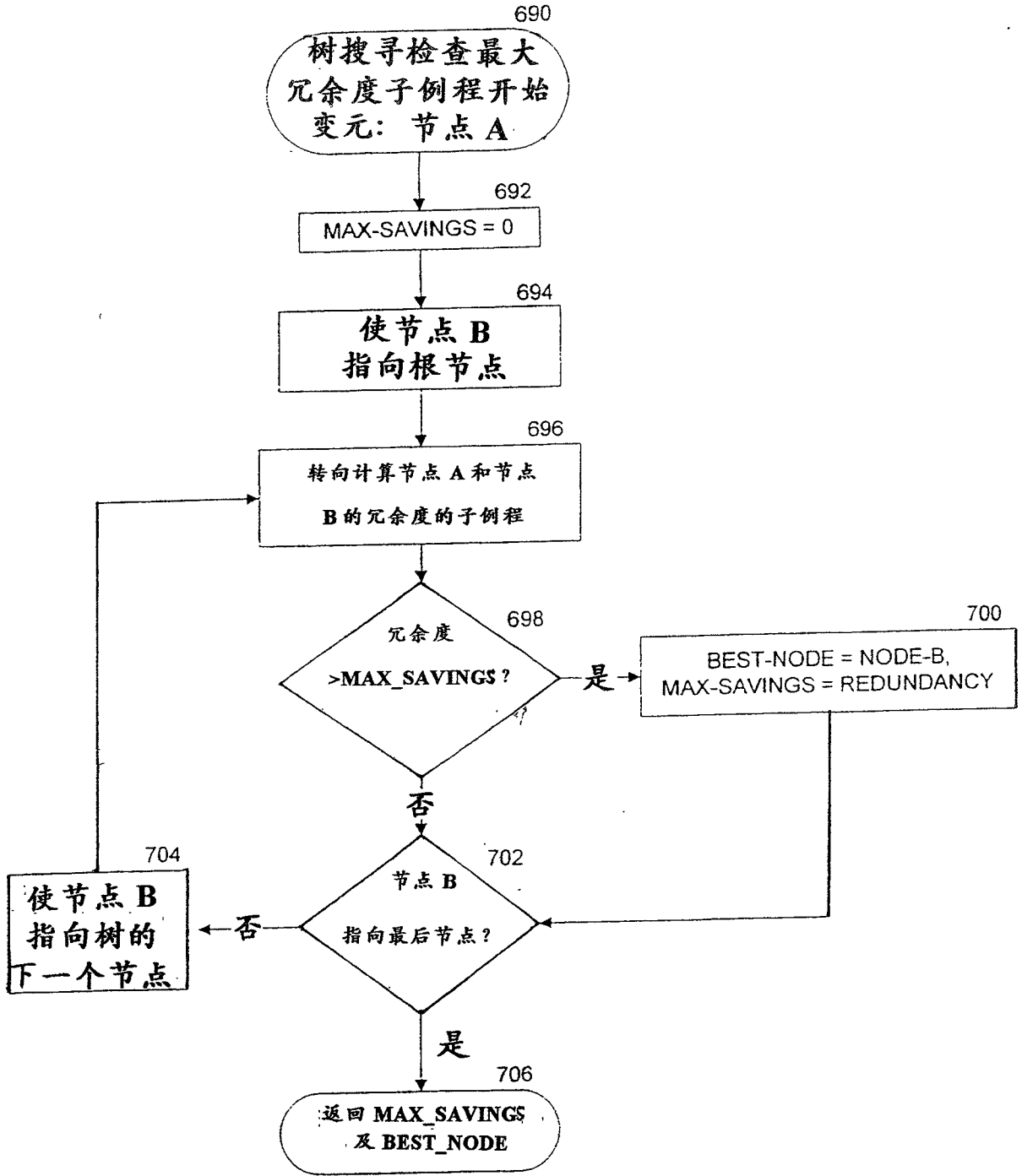


图16

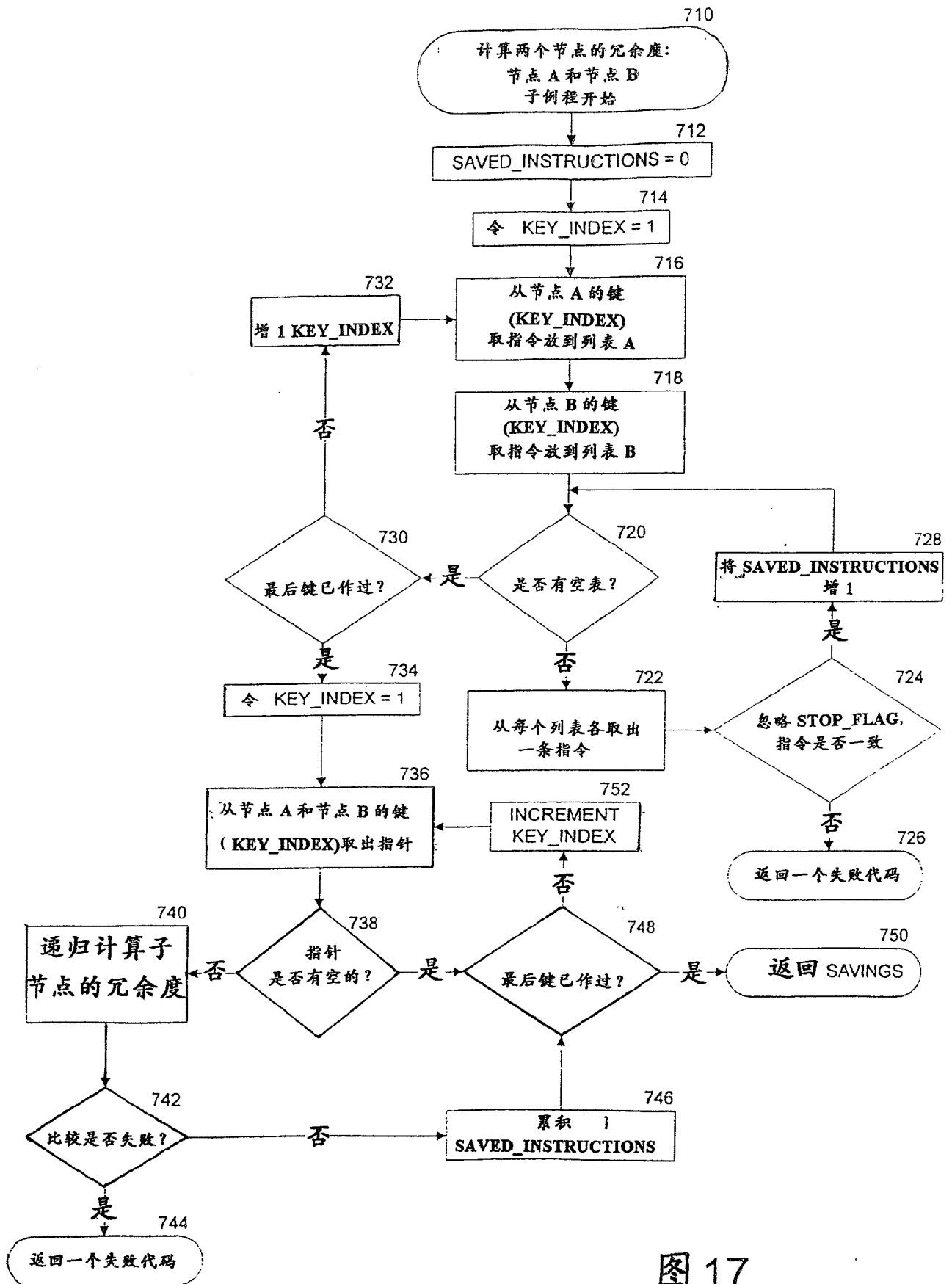


图 17



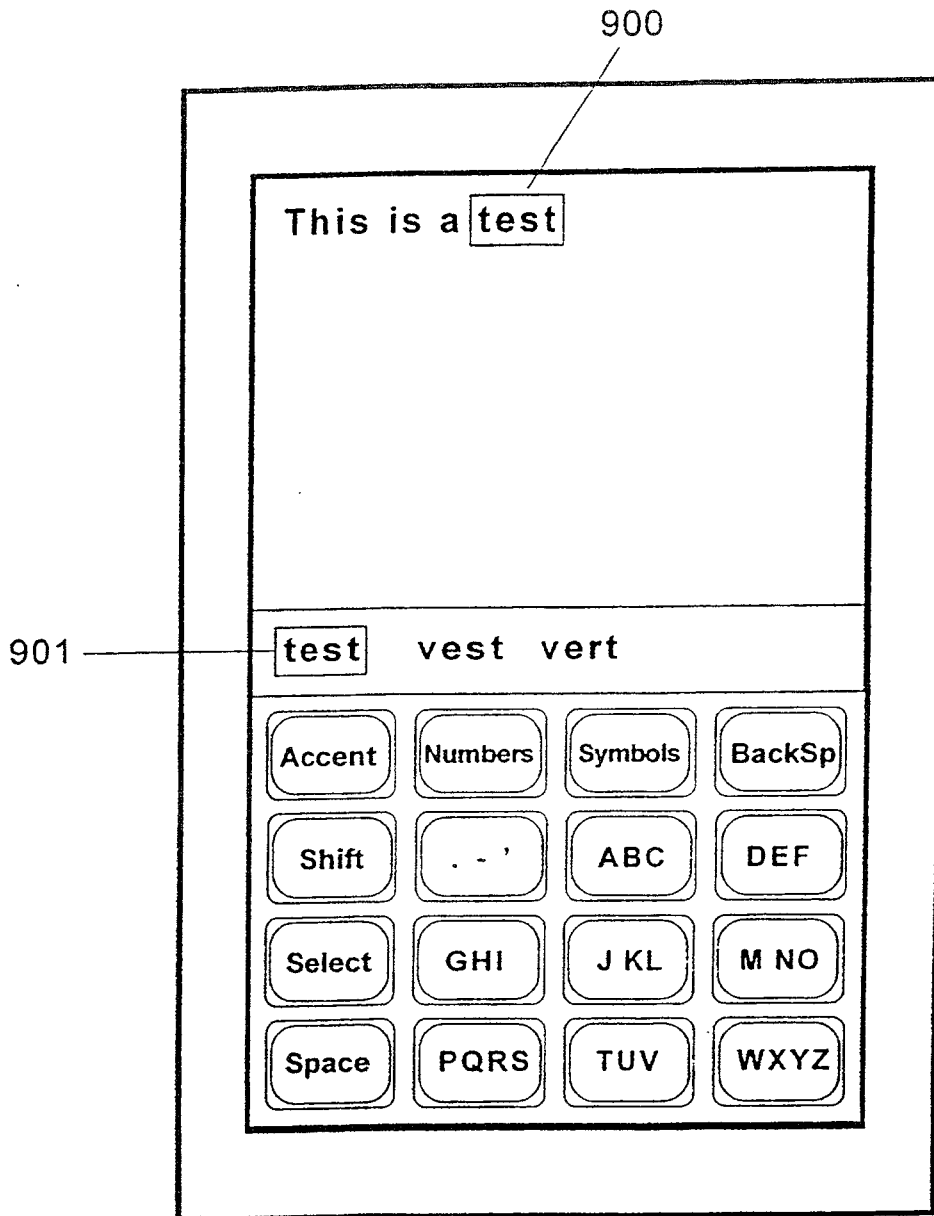


图 18A

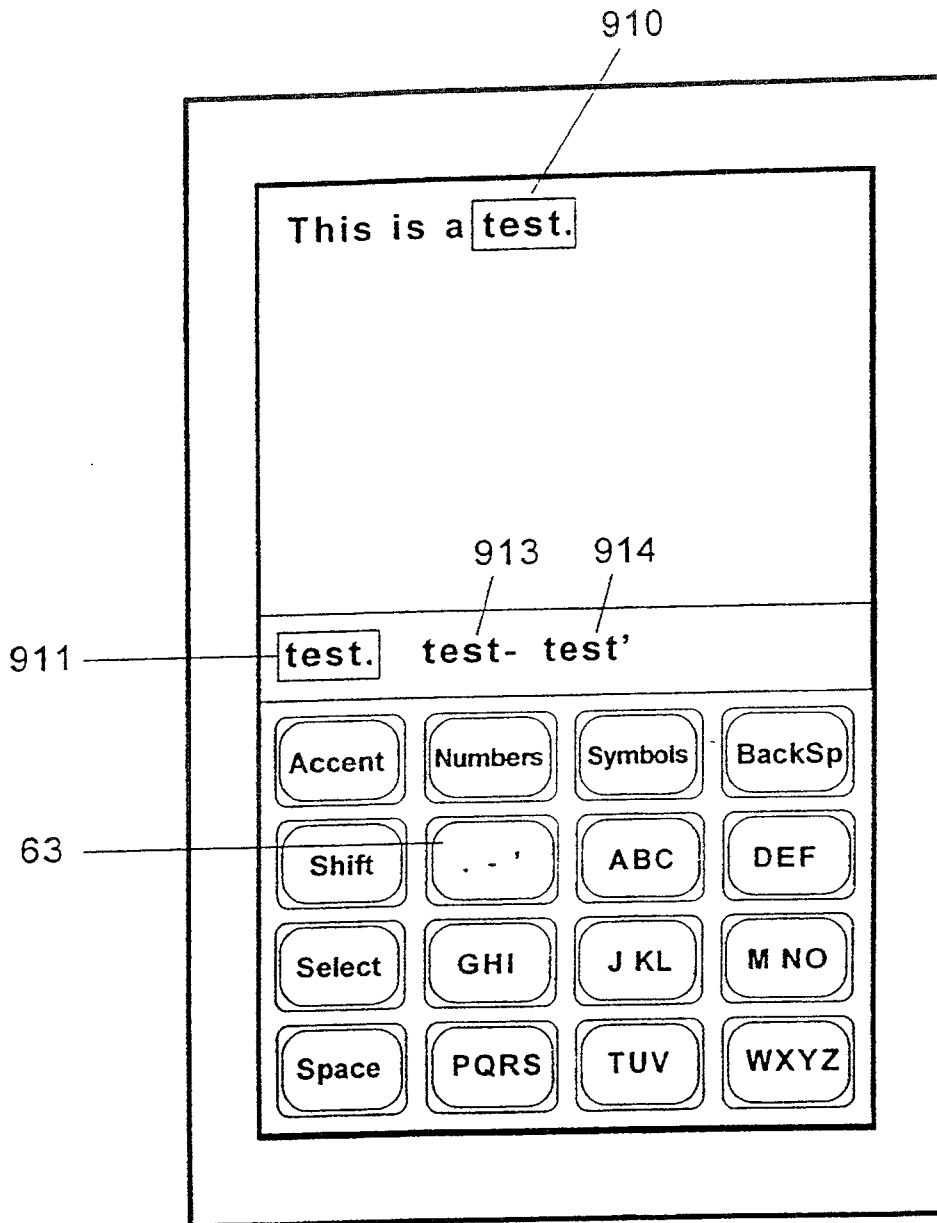


图 18B

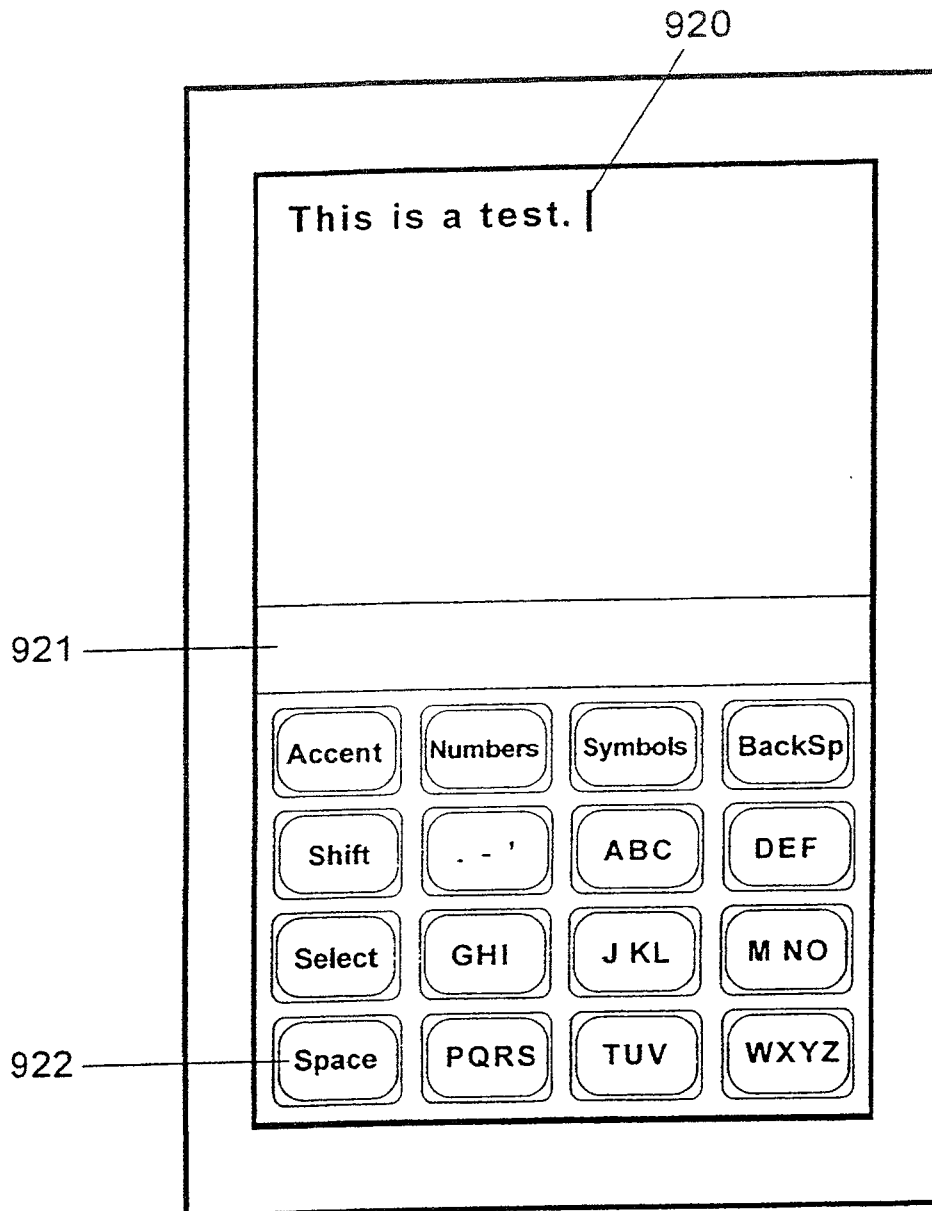


图 18C

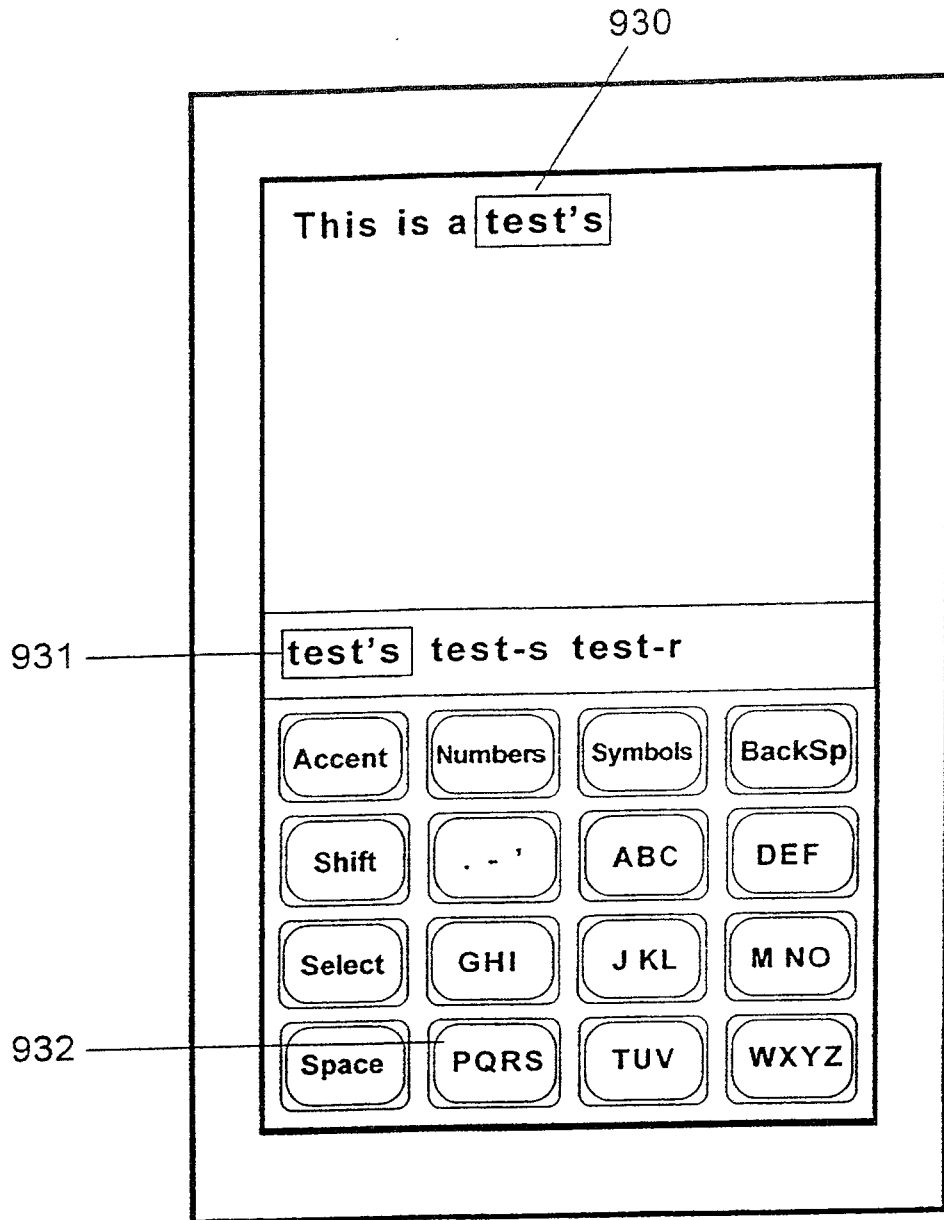


图 18D

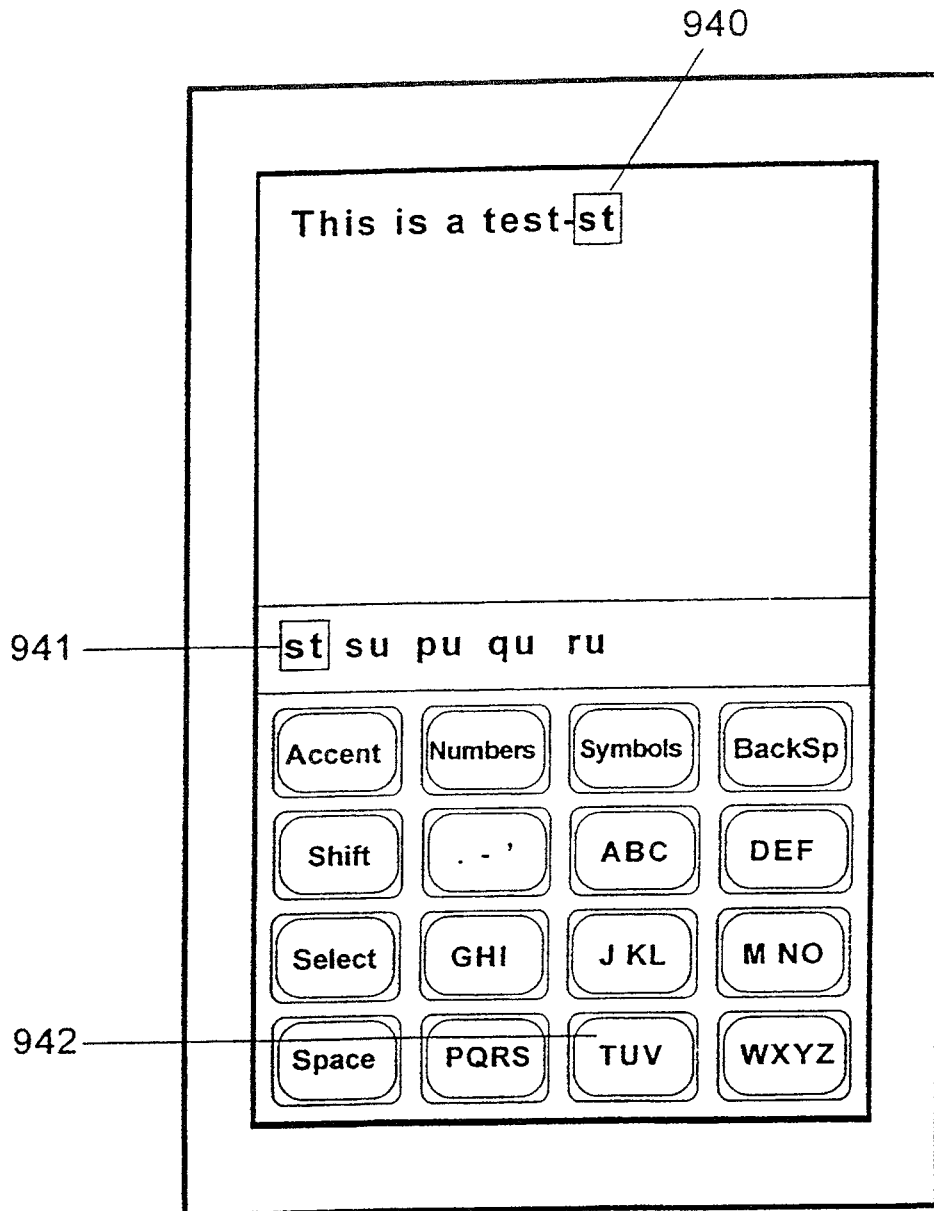


图 18E

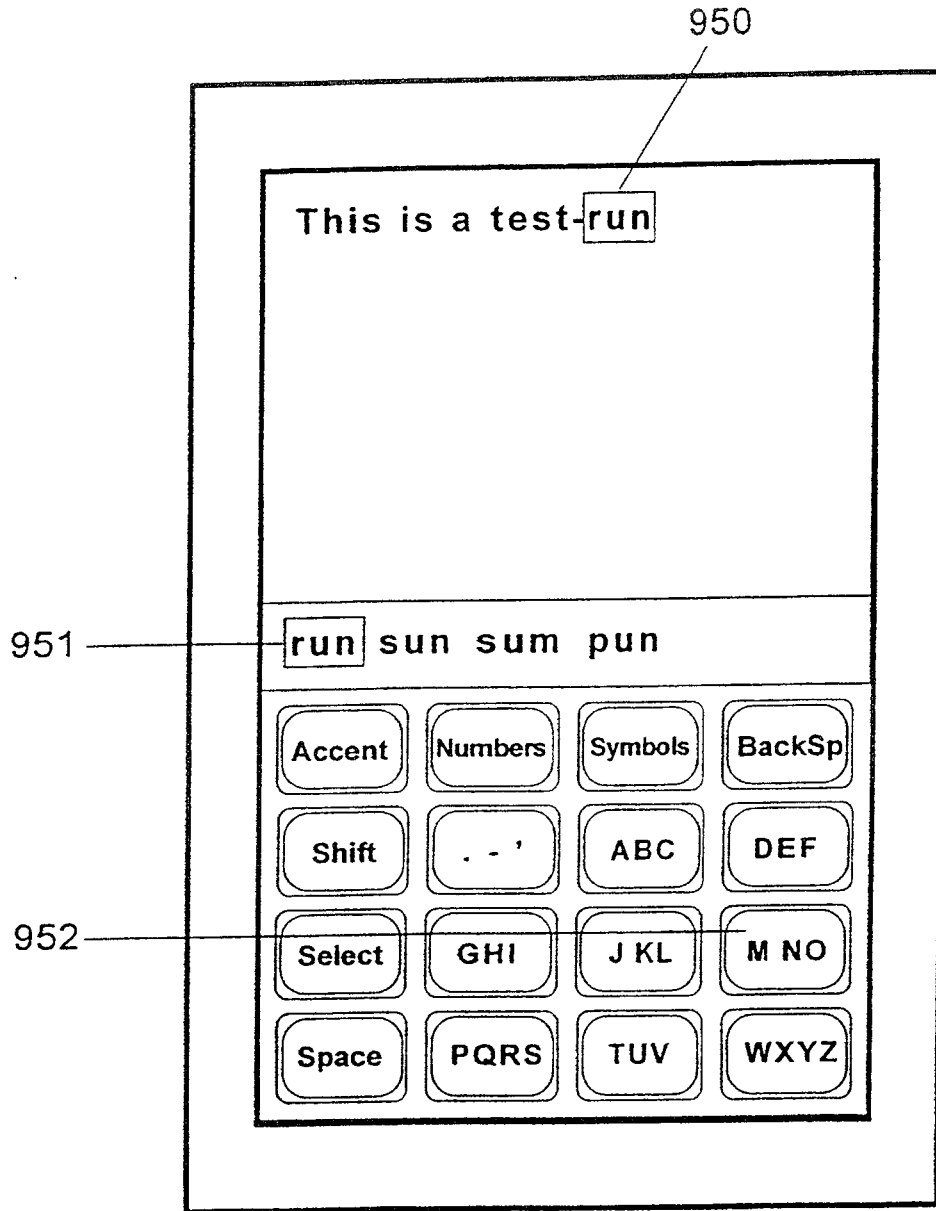


图 18F

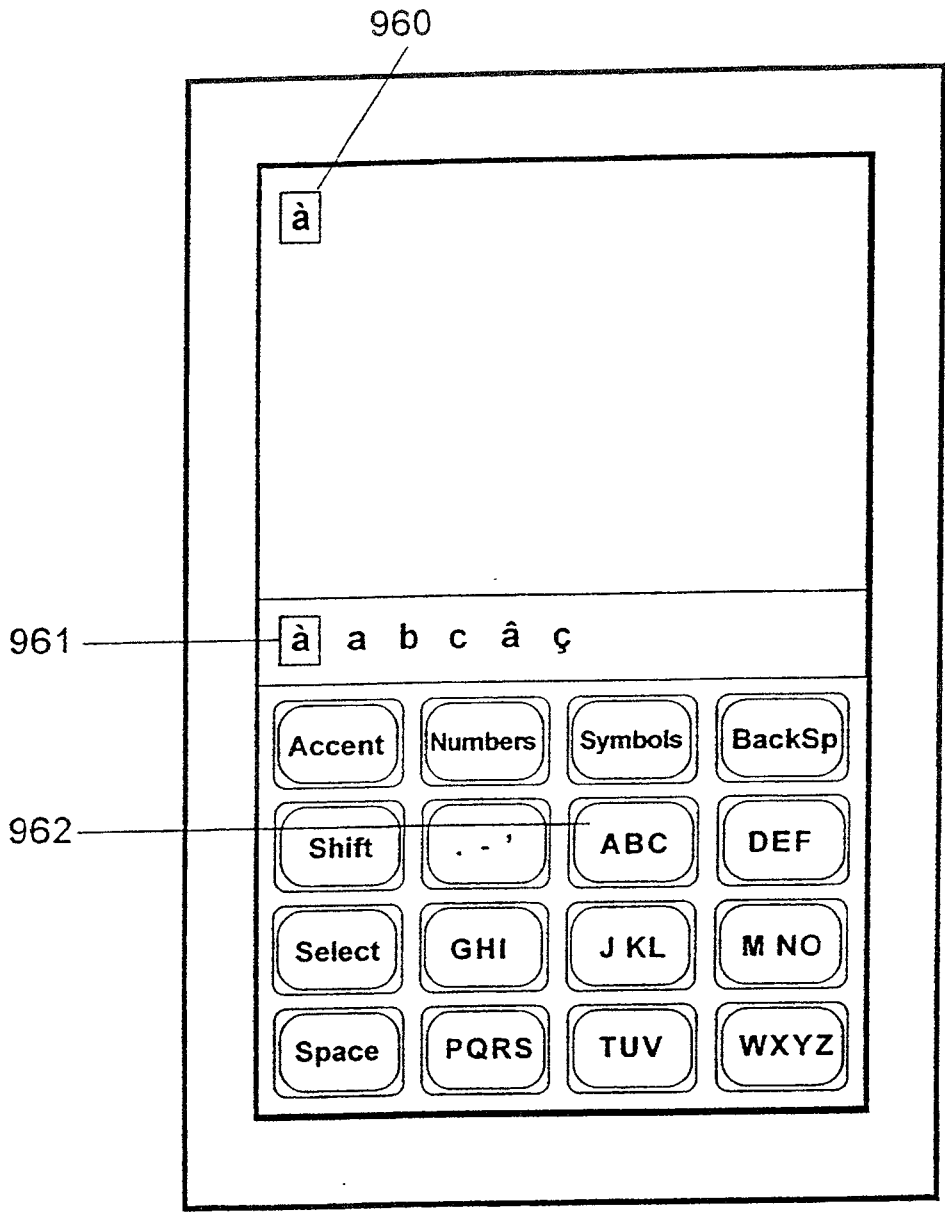


图 18G

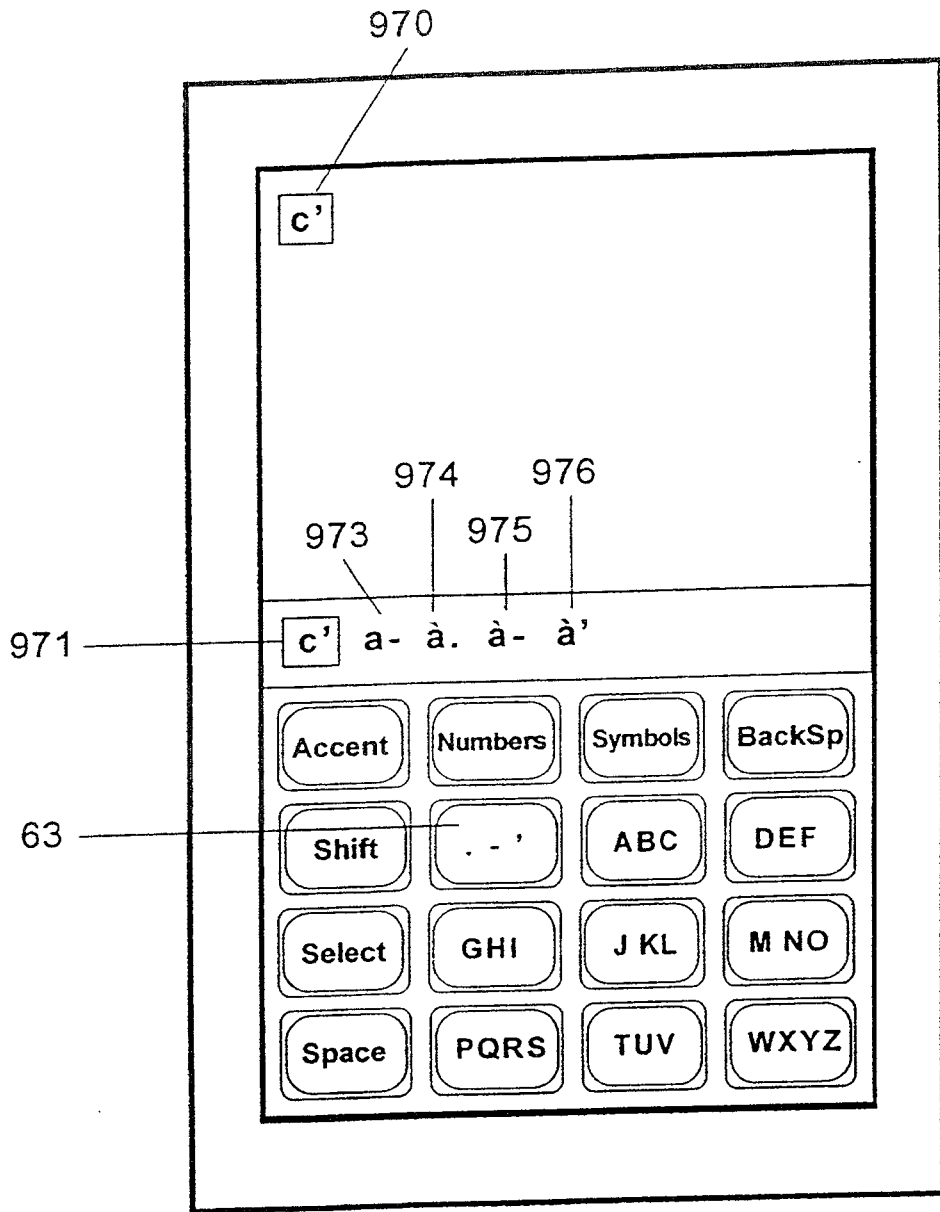


图 18H



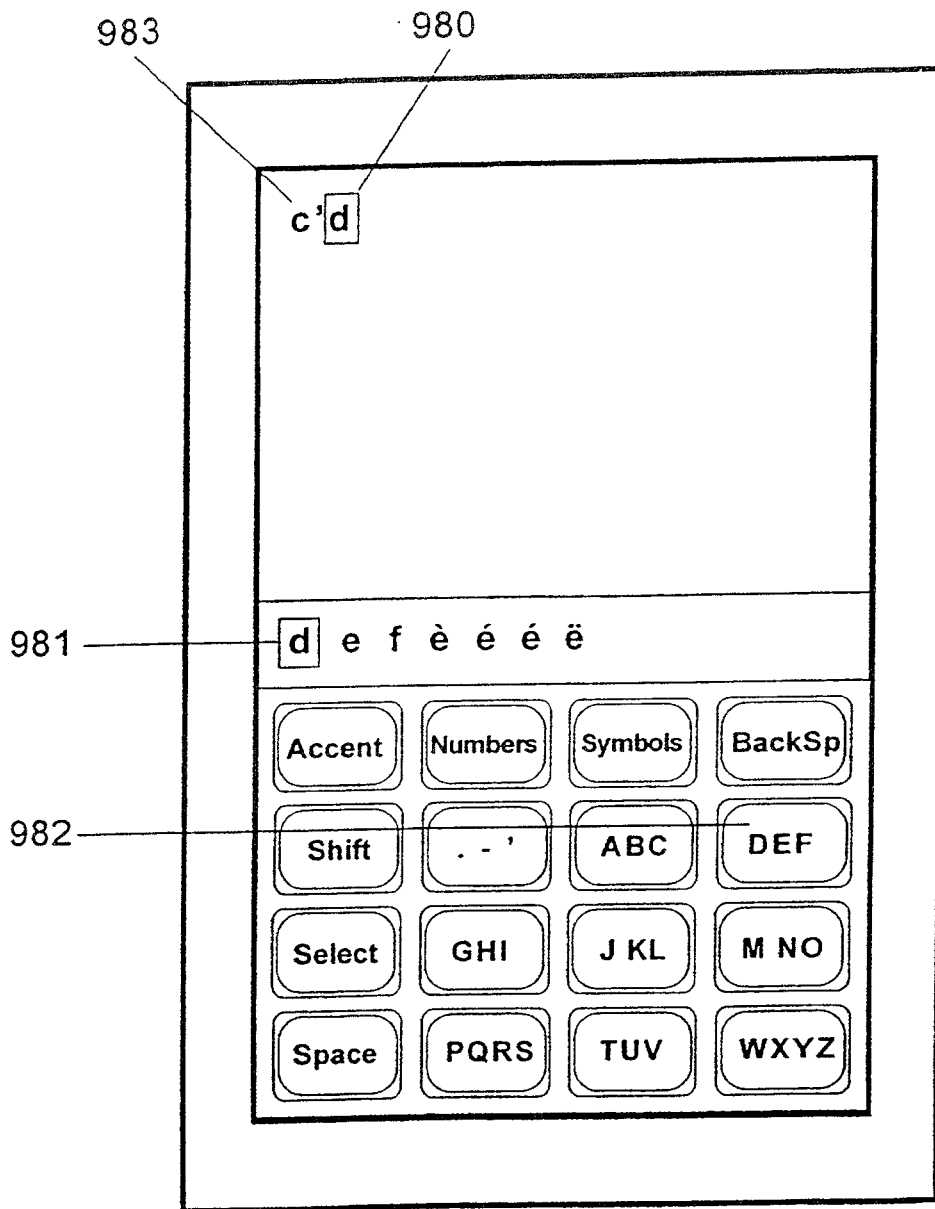


图 181

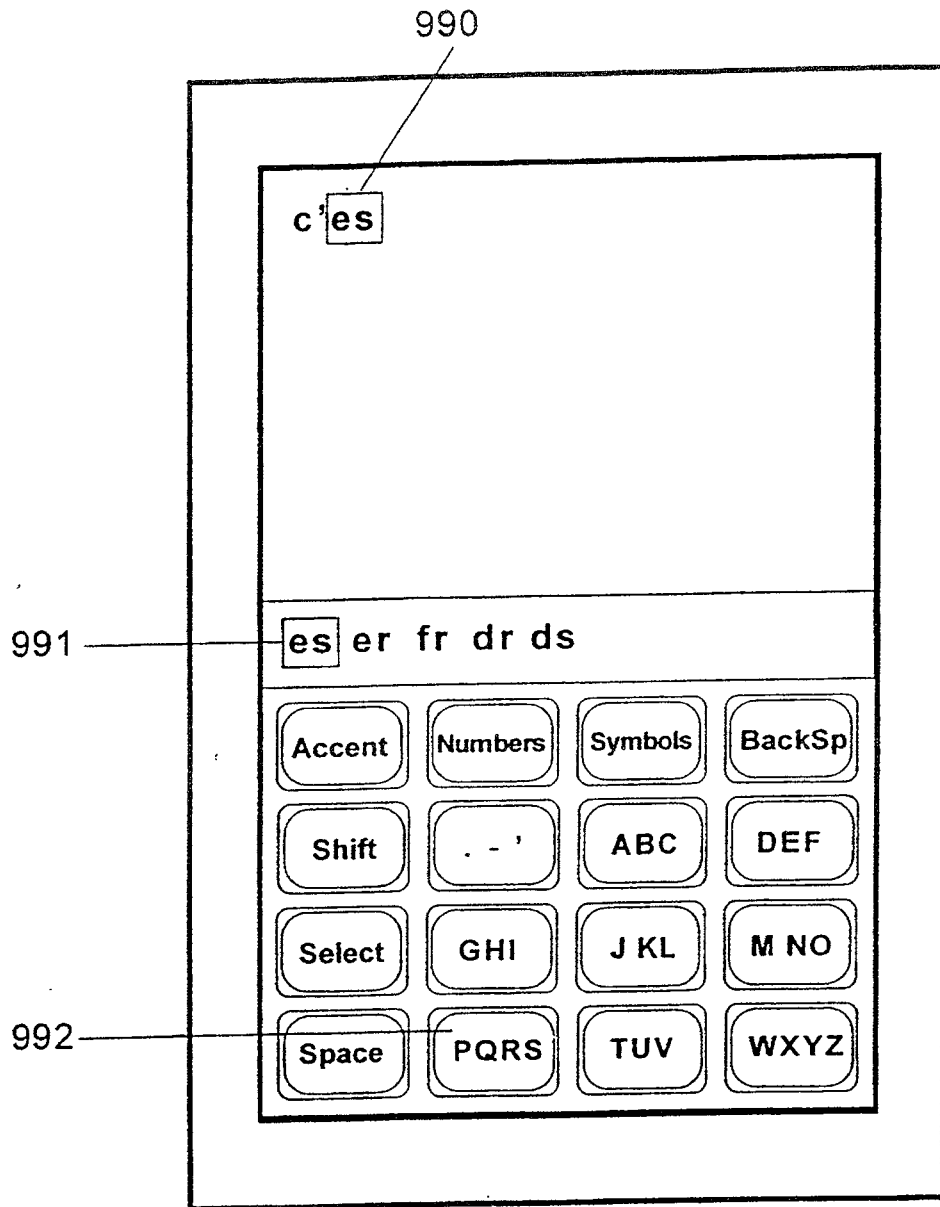


图18J

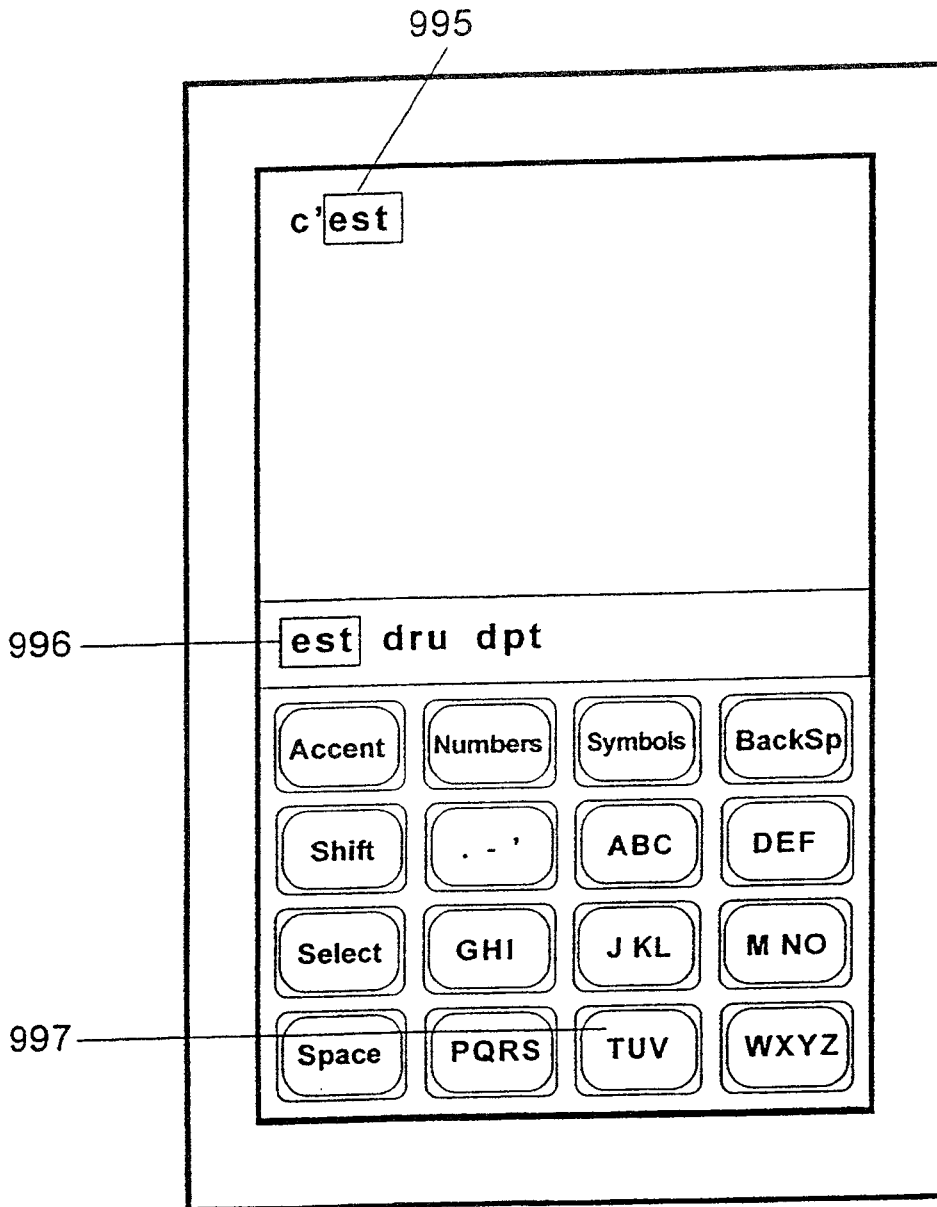


图 18K